

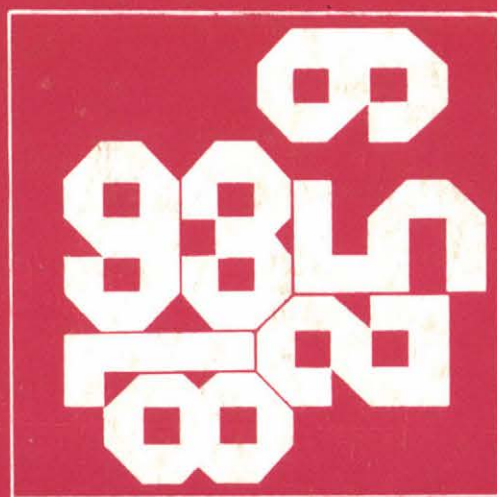
tanulmányok

67/1977

1977 MAR 06



MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ ITNÉZETE

AZ ANSI/X3/SPARC BIZOTTSÁG MODELLJE
ADATBÁZIS KEZELŐ RENDSZEREKRE

Írta:

GYÚRKI JÓZSEF

Tanulmányok 67/1977.

ALTERNATÍVÁK A TUDOMÁNYOS AKADÉMIA
TUDOMÁNYOS ÉS SZAKMAI KÖZVETLEN KÖZVETLEN

A kiadásért felelős:

DR VÁMOS TIBOR

ISBN 963 311 050 5

ISSN 0324-2951

TARTALOMJEGYZÉK

BEVEZETÉS	5
I. A SZABVÁNYOSÍTÁS TÁRGYA ÉS A MODELL ÁLTALÁNOS LEIRÁSA	6
1.1 A szabványosítás körülményei, előzményei, állása	6
1.2 A szabványosítás tárgya	9
1.3 Az adatbázis rendszer dinamikus működése általánosan	21
II. ELVEK, MODELLEK, FOGALMAK, DEFINICIÓK	27
2.1 Az érdeklődés birodalmi és az adatbázis	27
2.2 Általános használatu fogalmak	29
2.2.1 Objektumok fogalmai valamennyi adatbirodalomban	29
2.2.2 Kapcsolatok és kollekciók	37
2.2.3 Generikus objektumok	43
2.3 A valós világ	45
2.4 Koncepcionális modell	47
2.5 Externális modell	54
2.6 Internális modell	60
2.7 Adatmodellek és birodalmak kapcsolata	67
2.8 Objektumok kötése és leképzése egymás között	70
2.9 Adatfüggetlenség	73
2.9.1 Az adatfüggetlenség jellemzése	73
2.9.2 Az adatfüggetlenség specifikálása	81
2.9.3 Figyelembe veendő tényezők az adatfüggetlenségnél	84
2.10 Adatszótár/adat tartalomjegyzék	87
2.11 Változások és változtatások az adatbázis környezetében	91
2.11.1 A változások okai	92
2.11.2 A változások fajtái	97
2.11.3 Alkalmazkodás a változásokhoz	101
2.11.4 A változáshoz való alkalmazkodás gazdaságossága	104

2.12 (Humán) résztvevők az AEKR-ben	105
2.12.1 A résztvevők szerepe	106
2.12.2 A résztvevők jellemzése	114
III. AZ ADATBÁZIS RENDSZER MŰKÖDÉSE AZ INFORMÁCIÓS	
RENDSZER KÖRNYEZETÉBEN	119
3.1 Sémák kialakítása	120
3.1.1 A koncepcionális séma előkészítése	120
3.1.2 Az internális séma előkészítése	123
3.1.3 Externális sémák előkészítése	125
3.1.4 Leképezések a koncepcionális, internális és externális sémák között	127
3.2 Program fejlesztés	129
3.2.1 Program előkészítés	129
3.2.2 Forrás program-tárgy program konverzió	131
3.2.3 Végrehajtáshoz való előkészítés	133
3.3 Program végrehajtás	133
3.4 Felhasználói interface-ok	137
3.5 Az adatbázis fejlődése és karbantartása	137
IRODALOM	141

BEVEZETÉS

A jelen tanulmány az American National Standard Institute (ANSI) Standards Planning and Requirements Committee (SPARC) adatbázis munkacsoportjának 1975-ös riportja alapján készült. Ez a riport kis módosításokkal az ISO/TC 97 munkaanyaga is, mivel az USA hivatalos betérjesztéseként nemzetközi szabványosítás tárgyát is képezi.

A riport általános, elvi részének magyar nyelvű közreadásával az volt a célunk, hogy minél szélesebb körben népszerűsítsük, ismertessük a jelenleg legperspektivikusabbnak tűnő és szabványosítás tárgyát képező adatbázis rendszer modellt. Az említett riport tematikai strukturájától eltérünk, mivel célszerű volt bizonyos témakörök jobb megértése céljából az azonos problémakörökre vonatkozó (és riportban helyenként több fejezetre szétosztott) részeket együtt bemutatni. Bizonyos pontokon kiegészítéseket, magyarázatokat is fűzünk az anyaghoz. Ahol lehetséges volt magyar terminológiát használtunk, amely a vonatkozó angol nyelvű szakkifejezések értelemszerű fordítása.

A jelen tanulmány az "ANSI/X3/SPARC Interim Report 1975" anyag I-II-III és VIII. fejezetét foglalja magába.

A IV-V-VI-VII fejezetek anyagát később sorra következő közleményben szándékozzuk közreadni. Ezek a fejezetek a kijelölt inter face-k részletes leírását (IV. fejezet), az adatbiztonság (V. fejezet), adatintegritás (VI. fejezet) és a hibaállapotból való feléledés (VII. fejezet) kérdéseit tartalmazzák.

I. A SZABVÁNYOSÍTÁS TÁRGYA ÉS A MODELL ÁLTALÁNOS LEÍRÁSA

1.1 A szabványosítás körülményei, előzményei, állása

Bizonyos idő óta nyilvánvalóvá vált, hogy az ABKR információ feldolgozó rendszerek központi elemének tekinthető és nincs teljes egyetértés ezek tervezése, implementálása tekintetében. Annak ellenére, hogy számos implementáció létezik ilyen rendszerekre, továbbá létezik néhány dokumentum, amelyeket az információ feldolgozó szakemberek bizonyos csoportjainak képviselői hoztak létre közös erőfeszítéssel részben javaslatként speciális rendszer architektúra formájában (CODASYL, Codd), részben követelmények felsorolásaként (GUIDE-SHARE, CMSAG), nincs egyértelműen követhető, megalapozott irány a felhasználó, vagy rendszer fejlesztő részére. Vita folyik arról, hogy a létező implementációk, illetve a kidolgozott javaslatok teljesítik-e az elvárt követelményeket, másrészt a felvázolt követelmények valamennyien szükségesek-e. A vitát bonyolítja, hogy súlyos ellentétek vannak a felállított követelmények elérhetősége, gazdaságossági kihatásainak becslése tekintetében is.

Ezekben túl vita folyik a használandó adatmodell jellegéről: relációs, hierarchikus, hálózat. Lassan teológiai vitának is lehet ezt nevezni, amit személyes, üzleti és presztizs kérdések bonyolítanak és árnyalnak. 1972 őszén világosan felismerhető volt, hogy a növekvő zavart újabb szabályozott akcióval kellene fékezni és tisztázni. Ezért az ANSI Standard Planning and Requirements Committee (SPARC) Computer and Information Processing szekciója (X3), amelynek feladatai közé tartozik e területen ajánlások létrehozása az anya bizottság akcióit serkentendő (a területre vonatkozó szabványok és vezérvonalak kidolgozásában), létrehozott egy ad-hoc bizottságot az ABKR potenciális szabványosításának tárgyában. A bizottság

feladata volt, hogy meghatározza - ha van egyáltalán - a jelenleg szabványosításra érett témákat az ABKR témakörben az USA-ban. Az "egyáltalán" kifejezés fontos, mert a negatív válasz is fontos eredmény a szabványosítás szempontjából. A "jelenleg" kifejezés szintén fontos, jelezve, hogy a követelmények, technológiák és a gazdaságosság körülményei állandóan módosulnak, tehát folyamatosan kell újraértékelni őket az átváltási idő felismerése szempontjából.

A javaslatokban kifejtett szempontokkal az volt a cél, hogy alkalmas alapot teremtsenek a szükségszerűen előálló változások, valamint a tárolási, keresési technológiák fejlődésének követésére szervezeti egységeken (vállalat) belül és az adatfeldolgozási környezetben. Ezt elsősorban az Internális és Externális sémák közötti leképzések generálásával, működésük irányításával kívánják elérni, ami adatfüggetlenséget biztosít és a vállalat adatbázisának rendezett átszervezését teszi lehetővé. A használandó adatmodellek kérdésében biztosítani szeretnék, hogy olyan mechanizmus, implementációs spektrum legyen elérhető, amely valamennyi, szemléletében eltérő adatmodellre igaz. (Az adatmodellek egymásba történő átvitele, transzformációja a bizottság munkáján kívül is vizsgálat tárgyát képezi és jelentős közeledés tapasztalható a kezdetben elmentétes elvekben).

Fontos mellékterméke e munkának az ABKR-ekkel szembeni követelményhalmaz kifejlesztése. Mivel egyetlen létező vagy javasolt implementáció sem elégíti ki a teljes felhasználói spektrum követelményeit, helyes az elveket széles körben kifejteni, a javasolt adatmodelleket diszkutálni, implementációs kérdéseket analizálni. A bizottság munkájának eredménye riport sorozat lesz, melyben kijelöli a potenciálisan szabványosítható elemeket az ABKR területen és ajánlja, hogy van-e szükség, technológiai megvalósíthatóság és gazdaságossági igazolás szabványo-

sitási projekt kezdeményezésére. (Az első vizsgált interface a 7-es jelű a COBOL nyelv szempontjából. Ezen munka befejezésének tervezett ideje 1976 eleje. Belső riportként dokumentumot hoztak létre, ami széles körben publikálásra került.)

részben ezen munka hatására az ISO is hasonló akciót indított. Az ISO/TC 97 8. Plenáris ülésén (Program nyelvek) az 5. munkacsoporthoz utalta az adatbázis kezeléssel kapcsolatos krédésköröket. Így az SC 5-öt utasították, hogy hozzon létre munkacsoportot e témakörben. Több ország nyújtott be anyagot, az USA részéről a "SPARC Interim Report 1975" került beterjesztésre. A munkacsoport 1975 június 24-26 között Washingtonban ülésezett, ahol állást foglalt a benyújtott anyagokkal kapcsolatban és meghatározta a nemzetközi szabványosítás érdekében teendő további lépéseket. (A munkacsoport következő ülése 1976 január 12-15 között Párizsban volt.)

Az ISO/TC97/SC 5 megállapításai a Washington-i ülésen a következők voltak:

1. A munkacsoport arra a megállapításra jutott, főleg Hollandia (ISO/TC97/598) javaslata alapján, hogy bármilyen szabványosítási akció az ABKR területén a jelenleg létező javaslatok alapján időelőtti. Ez első sorban azon kritériumok hiányának tudható be, amelyek szerint értékelhetnénk a létező javaslatokat.
2. Az ANSI/X3/SPARC Interim Report (ISO/TC97/SC5 (USA-75) N359) elfogadható a diszkusszió kiinduló bázisaként az ABKR-ek átfogó architektúrája szempontjából.
3. A munkacsoport fontosnak tartja és szükségesnek érzi, hogy az ABKR mindenfajta felhasználóját azonosítsuk és specifikáljuk követelményeiket, elvárásaikat az ABKR-rel szemben.

4. A munkacsoport javasolja az N359 riport terminológiai részének és a benne foglalt általános elveknek kiegészítését. Kezdeti erőfeszítésként a munkacsoport prioritást ad az N359 riportban kiemelt interface-oknak és azok további vizsgálatát javasolja. Ezeket a prioritásokat a szabványosításból származó előnyök optimalizálásához használhatjuk fel.
5. A fenti tevékenységekkel párhuzamosan a CODASYL adatbázis tevékenységét is értékelni kell. A munkacsoportnak az a véleménye - a különböző nemzeti szervezetekben folyó tevékenységek áttekintése alapján - , hogy a CODASYL által készített specifikáció jelenlegi formájában nem alkalmas szabványosításra.
6. A munkacsoport kutatás/fejlesztési munkát ajánl azokhoz az interface specifikációkhoz, amelyek érettek szabványosításhoz és amelyekhez megfelelő javaslat-jelöltek kidolgozás alatt vannak.

1.2 A szabványosítás tárgya

A SPARC munkacsoport kezdeti feladata a különböző nézetek megértése és respektálása volt, mivel valamennyi alternatív adatmodell képviselője jelen van a bizottságban. Létrehoztak egy terminológia gyűjteményt, ami konzisztens és kölcsönösen átfogó. Valószínű, hogy ennek további bővítése, finomítása szükségessé válik, de jelentős közeledés volt tapasztalható a terminológiák egyeztetése révén is.

A kezdeti diszkussziókon kellett eldönteni, hogy mit tekintsenek az ABKR tárgyának és azon belül mi legyen kezdetben (és egyáltalán) a szabványosítás tárgya. Így vizsgálat alá vették az információs rendszerek öt fő komponensét (általuk alkalmazott felbontás), nevezetesen:

1. Üzenetek
2. Rekordok
3. Eljárások
4. Erőforrások
5. Folyamatok

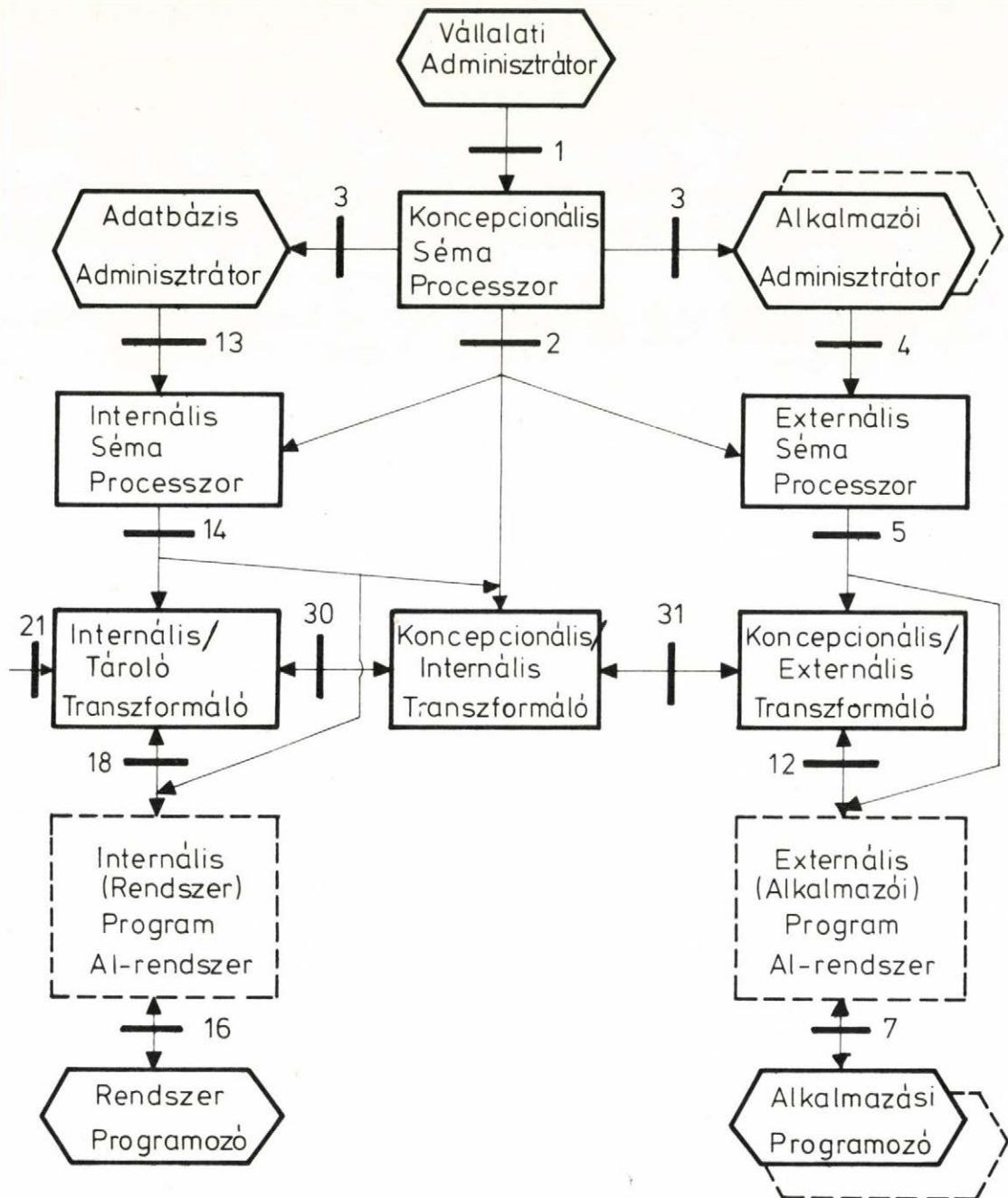
Az értékelés a következő eredményt hozta:

- 1) A valóságos Bemenet/Kimenet (kártya bemenet, printer kimenet, terminál I/O) és a folyamatok között átvitelre kerülő adatok üzeneteknek tekinthetők és az "üzenet kezelés" fogalma alá foglalhatók, ami nem része az ABKR-nek.
- 2) Minden tevékenység, ami programok előkészítésével, fordításával, ellenőrzésével, katalogizálásával, stb. kapcsolatos és annak érdekében történik, hogy futóképes tárgyprogramokat kapjunk, az "eljárás kezelés" alá tartozik és mint ilyen nem része az ABKR-nek.
- 3) A memória allokálás, spooling/swapping/ dispatching, diszk és a szalag egység hozzárendelések valamint egyéb teendők a számítógép erőforrásainak kezelése terén "erőforrás kezelés"-nek tekinthetők, ami nem része az ABKR-nek.
- 4) A lokális változók, munkaterületek, utasítás számlálók kezelése, továbbá egyéb teendők a folyamatok állapotával kapcsolatosan "folyamat kezelést" jelentenek és ez sem része az ABKR-nek.
- 5) Így a bizottság vizsgálat tárgyának tekinti a rekordokat, mezőket, file-okat, halmazokat és ezen adatobjektumok leírásait, az indexeket, leképezési technikákat, hozzáférési módokat, file szervezéseket és felhasználói nyelveket, amiket összefoglalva az ABKR fogalma alá von.

Ugyanakkor hangsúlyozni kell, hogy az ABKR alá nem tartozó problémaköröknek szoros kapcsolatuk van a kifejlesztési kívánt adatbázis modellekkel és a javasolt ABKR mechanizmusával, dinamikus működésével, mivel egy átfogóbb rendszerben (információs rendszer vállalatra, szervezetre) élnek egymás mellett. Ez a tény, továbbá az ABKR területen belül folytatott vizsgálat a szabványosításra szóbajövő elemek detektálása érdekében vezetett oda, hogy azt, amit a szabványosítás egyáltalán érinthet, az *interface*-ek alkotják. Nincs értelme olyan szabványnak, ami specifikálja hogyan kell a komponenseknek, részrendszereknek működni. Az való szabványosításra, ahogyan a komponensek összeilleszthetők, vagyis interface felületeik. Ezért olyan általánosított ABKR modell kifejlesztését akarják elérni, ami kiemeli az interface-eket és azokat az információ/adat féléseket, amelyek rajtuk áthaladnak a rendszer működése során. Ezen adatbázis kezelő rendszer modellnek, két változata van: egy egyszerűsített (1.1 ábra) és egy teljesebb, (1.2 ábra) amelyekben az interface felületeket számokkal jelölték a könnyebb és egyértelműbb hivatkozás céljából. (Az egyszerűbb modellben a részletesebb képpel való összehang érdekében ugyanazon interface számozást követték, ezért ott bizonyos számok hiányoznak.)

A két modell közötti alrendszer elhagyások, összevonások mutatnak bizonyos lehetőségeket, ahogyan a rendszer programozók, gép operátorok megközelítik, értelmezhetik a rendszert, implementáláskor ad-hoc javításokat téve.

A rendszer működési mechanizmusából bizonyos konverziós utakat lerövidíthetnek, elhagyhatnak, hatásosabb működés biztosítása érdekében. Ugyanakkor vitatható ezek célszerűsége az adatfüggetlenségre, adatintegritásra és a biztonság megoldására való kihatásuk, valamint az adatnak adott tároló közegen fizikai struktúrába való leképezése bonyolultsága szempontjából. A tárolási struktúra kérdéseivel (amelyek a 21-es interface-től balra találhatók) a munkacsoport egyáltalán nem foglalkozik, mivel ezeket nagy-



1.1 ábra

részt a jelenlegi tároló közegek fizikai törvényszerűségei diktálják és kevésbé stabilak. A "változások"-hoz való alkalmazkodó képességet az ABKR-ben egyrészt az e területen mutatkozó állandó fejlődés kívánja meg, tehát szabványosításra érett dolgokat itt nem célszerű keresni.

Az ABKR modellben kijelölt interface-ok technológiai karakterét - az ember-gép interface-ket kivéve - a kidolgozott javaslatok nem határozták meg. Azok lehetnek hardware-ban, software-ban megoldva, speciális egyedi, vagy keverék megoldások egyaránt elképzelhetők. Ugy szintén nincs előírva a rendszer implementálásának módja sem, csak a teljesítendő követelmények vannak specifikálva. Több helyen azonban, ismerve az egyes adatmodellek (relációk, hálózat) implementálásainak tapasztalatait, utalnak a legpraktikusabb realizációra.

Mindkét ABKR sémában jól látható a SPARC riport leglényegesebb aspektusa, nevezetesen a 3 szintű (vagy 3 sémás) megközelítés az adatbázis objektumainak leírásánál és az erre épülő adatbázis kezelési mechanizmus. A 3 "adatbirodalom"-ra vonatkozó információ tulajdonságai, formája erősen eltérhet, ezért adatmodelljeik és sémájuk különböző objektumok tárgyasulása is eltérő, amit az ABKR működésének dinamikája határoz meg.

Jelentős aspektus továbbá a négy emberi szerepkör elkülönítése, vagyis a vállalati, adatbázis és alkalmazói adminisztrátorok valamint az alkalmazói programozók megkülönböztetése. (A rendszer programozók az adatbázis és az ABKR normál működéséhez "externális" személyek, ezért kezelik őket az előbbi négytől eltérően.) Ugyanaz a személy lehet több szerepben is egyidejűen vagy egy szerep több személy között lehet felosztva. Fontos, hogy egyetlen vállalati és adatbázis adminisztrátor van szemben a másik két csoporttal, ahol több személy létezhet. Ez vezet ahhoz, hogy lehet több externális séma, mindegyik másféle adatszemplélettel, feltéve hogy azok konzisztensek

és származtathatók az egyetlen koncepcionális sémából. Természetesen több alkalmazói programozó használhatja ugyanazt az externális sémát, bár nem dolgoznak valamilyen módon ugyanazon alkalmazói programon.

Minden adminisztrátor meghatározott területen felelős a rendszer számára szükséges adatszemplélet specifikálásáért, adatobjektumok és köztük fennálló adatközötti kapcsolatok leírása formájában. A központi, alapvető szemlélet a vállalati adminisztrátoré, aki a koncepcionális sémát definiálja. Hangsúlyozni kell ismételten, hogy a leggyakrabban elhanyagolt dolog információs rendszerben a koncepcionális séma explicitté tétele. Ez egy reális, felfogható, számítógép által olvasható formában megadható specifikáció, amit potenciálisan szabványosítható szintaxisu nyelven adhatunk meg. (Számos tanulmány foglalkozott a SPARC riport óta, ilyen nyelvekkel és a koncepcionális séma szemantikai tartalmával, amihez azonban szükséges a rendszer dinamikus működésének vizsgálata és ismerete.)

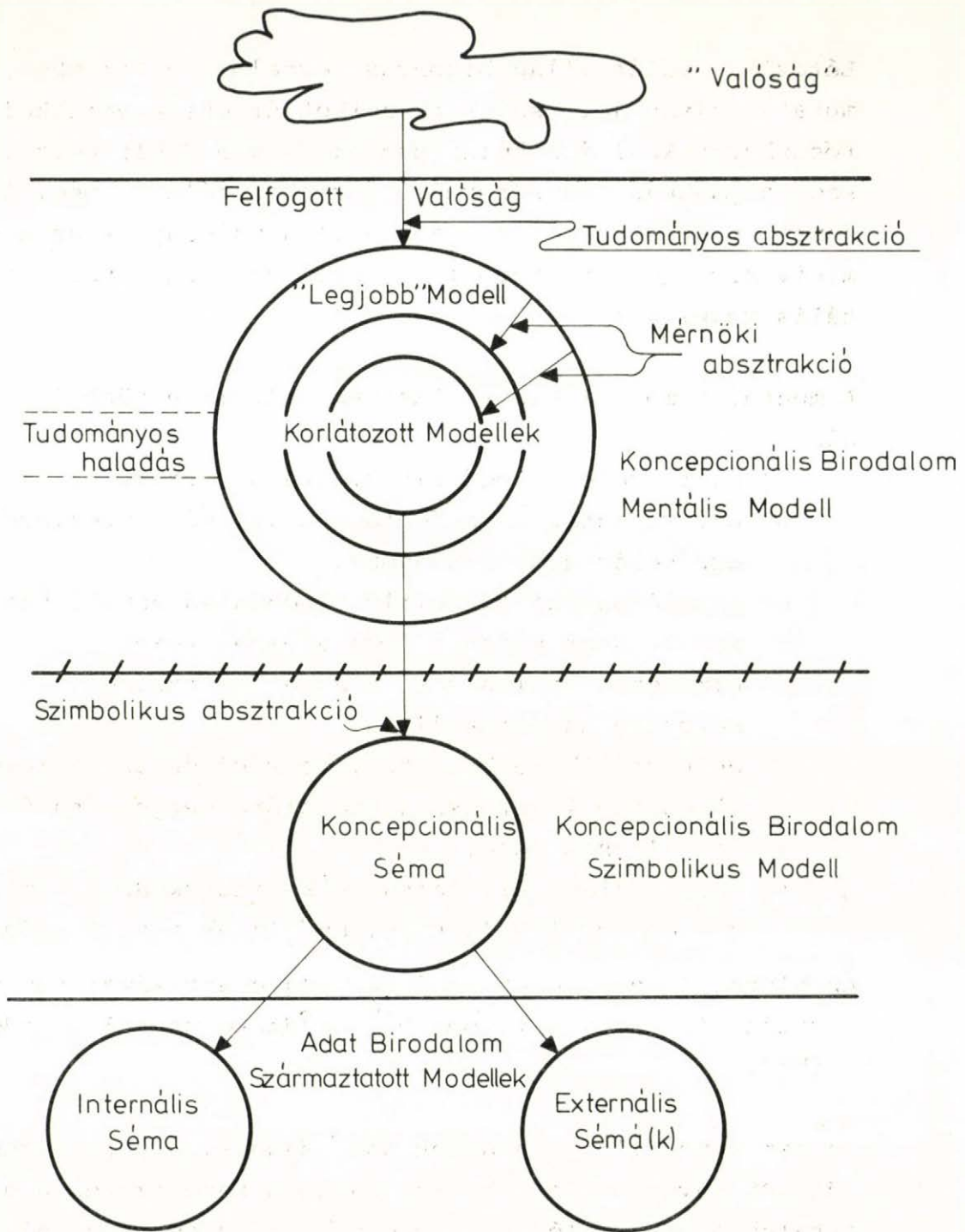
A koncepcionális séma bevezetésének előnyei az alábbiak:

- a) előkészítése azt eredményezi, hogy a vállalat formalizálni kénytelen információs modelljét. Ez a tevékenység színvonalas tervezési feladat, aminek eredménye stabil referencia az ABKR installálásához.
- b) ezen referencia alap birtokában az információs igények összegyűjthetők. Ezzel előáll az adat szótár/ adat tartalomjegyzék alapja a vállalatnál, megmutatva hol, hogyan, ki által használtatik információ
- c) a központi referencián túl ez a séma (és adminisztrátora) az a autoritás, amely biztonsági, integritási problémákat feloldhat. Ezzel a biztonsági hierarchia a koncepcionális sémában alakítható ki, ami átterjeszthető a többi sémára, biztosítva automatikusan, hogy a később kialakuló sémák biztonsági deklarációi egymással konzisztensek legyenek.

- d) ha m -számu internális sémánk és n -számu externális sémánk van, akkor az összes transzformáció típus $m \times n$ lenne. Ha azonban minden externális sémát előbb a koncepcionális sémába transzformálunk, továbbá az internális sémákat a koncepcionális sémából származtatjuk (és viszont), akkor csak $m+n$ leképzés megvalósítása válik szükségessé.
- e) a koncepcionális séma elszigeteli az externális sémákat (felhasználó adatszemlélete) az internális sémáktól (tároló szerinti adatszemlélet) a fenti transzformációs lehetőségek miatt. Így internális sémabeli változások és a tárolási technológia fejlődése csak a koncepcionális és internális séma közötti leképzést befolyásolja. Természetesen implementáció függő és gazdaságossági döntés eredménye a valóságos transzformációs mechanizmus milyensége.

A koncepcionális séma mögötti meghatározó gondolat az "entitás-tulajdonság-érték" hármas, amit a GUIDE-SHARE riportbeli követelmények tettek explicitté. A SPARC munkabizottság tagjai egyetértenek a koncepcionális sémával összefüggő elvekkel, de eltérőek a vélemények a teljes ABKR modelljében való helyét és fontosságát illetően. A modell dinamikus működésének mechanizmusába eltérő módon lehet beépíteni a koncepcionális sémát eltérő működési sajátosságokat (válaszidő, memória igény) kívánván elérni, de egyuttal más szintű adatfüggetlenséget, változásokhoz történő alkalmazkodó képességet kapva. A különvélemények határozott kifejtését, egyeztetését fontosnak tartja a munkabizottság, mert a későbbi fázisokban azok akadályozhatják a továbbhaladást.

A 1.1 és 1.2 ábrán vázolt adatmodellek származtatásának logikai gondolatmenetét követhetjük az 1.3 ábrán, ami a tudományos kifejtés, megismerés általános sémájának egy változata. Ezen követhető, hogyan jutunk el a



1.3. ábra

valóságtól azokhoz az adat modellekhez, amiket ténylegesen használunk alkalmazási programokban.

Létezik a valós világ bizonyos értelmes értelemben, felmutatva olyan jelenségeket, amiket érzékszerveinkkel mérőeszközeinkkel érzékelni tudunk. Így a "felfogott valóság" agyunkban transzformált formában jelenik meg. A tudományos absztrakció folyamatával a valóságnak ez a primitív képe áttranszformálható racionálisan a valóság mentális modelljébe.

A mentális modell kialakításának folyamata több lépésből áll:

- a) megfigyelés (meg/feljegyezve az érzékelést),
- b) experimentálás (a felfogott valóság ingerlése új érzékelés generálásához),
- c) általánosítás (intuitív elfogadása annak, hogy hasonló inger hasonló érzékeléshez vezet),
- d) elméletté fejlesztés (alapvető általánosítások keresése és detektálása),
- e) dedukció (következtetés, hogy új és az eddigiektől különböző ingerlés új, de előre megjósolható megfigyeléshez vezet),
- f) verifikálás (valóságosan alkalmazzuk az új ingereket és megfigyeljük, ellenőrizzük a várt effektust),

Az előbbi lépéssorozat ismételt végrehajtásával jutunk a valós világ egyre finomabb, pontosabb mentális modelljeihez.

Ha ezt a modellt át akarjuk adni másnak, a terjesztéshez közeget, nyelvet kell használnunk. A természetes nyelvek általában ben kielégítően precizek a tudományos modellek tartalmának kifejezéséhez. Helyettük ma a formális nyelvek használata látszik célszerűnek, bár vannak komplikációk a létező formalizmusok használata körül, ha a valóság tudományos leírását komprimálni akarjuk segítségükkel.

Igaz viszont, hogy a problémák nagyobb része a modellek határfelületein jelentkezik, ahol általában olyan mentális modellek érintkeznek, amelyek eltérő szimbolizmust kívánnak belső természetük miatt.

Általában nem törekszünk a valóság teljes és pontos modelljének meghatározására, a "legjobb" modell az, ami a tudományos megismerés eredménye lehet. Ennek határai állandóan mozognak a tudományos ismeretek módosulásával, bővülésével. Legtöbbször gyakorlati követelmény (fizikai korlátok), hogy a valóság egy részéről bizonyos aspektusairól korlátozott modellünk legyen, amit a "legjobb" modellből származtathatunk a "mérnöki absztrakció" folyamatával. Ezzel az illető alkalmazáshoz a valóság azon vonatkozásait tesszük hozzáférhetővé, amelyek ahhoz relevánsak és a többit elhagyjuk. Így a formális leírás mindig csak a megfelelő szintű absztrakcióval foglalkozik.

Ez az eredő formalizmus, ami "szimbolikus modell", korlátozott (mérnöki szemléletből származik) és tudatosan rész-halmaza a valóságnak. A szimbolikus absztrakció folyamatának eredményeként ölt formát, konvencionális, előre determinált szintaxist követ, forma halmazok, adat objektumok formájában, amelyekhez alkalmas szemantikai tartalom tartozik a hozzárendelési szabályok, igazság szabályok elfogadása, deklarálása következtében. Az adatfeldolgozással kapcsolatos valós világ (vállalat) modellezni kívánt ismert dolgainak összessége ez a modell, amit a koncepcionális sémával írunk le. Ebből a formális, szimbolikus modellből leképzéssel jönnek létre az internális sémák (egyidejűen csak egy) és az externális sémák a vonatkozó szakterületek specialistáinak közös szakismeretét ötvözve. A modell alkotásnak előbb vázolt folyamata nem specifikus az ABKR rendszerekre, de ezen folyamatnak egy szabványostíás akció keretében való tudatosítása fontos lehet annak garantálására, hogy a teljes modell működéséhez szükséges ismeretek (sémák paraméterei, adattartal-

muk) konzisztens folyamat eredményeként jöjjenek létre. Így a koncepcionális séma kialakítása és fontossága a SPARC riportban központi kérdés és feltételezik, hogy belőle direkt módszerrel (formális leképzéssel) származtatható valamennyi séma (hozzá viszonyítva alárendeltek).

Áttekintve ábrák és általános megjegyzések kíséretében a SPARC munkabizottság munkájának keretét és tartalmát, de nem fejtve ki nagy vonalakban sem a javasolt adatmodell dinamikus működését, érdemes konkrétan idézni hivatalos megbízásuk tárgyát és munkatervüket, hogy az esetleges hazai aktivitást ehhez igazíthassuk, vagy a hazai környezetben szükségesnek érzett kutató/fejlesztő munkát körülírassuk:

Tárgy:

Létező és a javasolt ABKR-nek áttekintése, olyan adatbázis kezeléssel kapcsolatos riportok és egyéb anyagok tanulmányozása, amelyek adatbázisrendszerekkel kapcsolatosak. Javaslatok kidolgozása (SPARC/90 dokumentumok) ezen a területen olyan részrendszerekre, amelyek alkalmasak vagy jelenleg alkalmatlanok Amerikai Nemzeti Szabványok (ANS) vagy vezérvonalak kifejlesztéséhez.

Munkaterv:

1. Információsrendszerek általános, átfogó strukturájának definiálása abból a célból, hogy azonosíthassuk az ABKR-hez tartozó részeket.
2. Az X3 és SPARC adatbázis rendszerek tárgyában végzendő tevékenységének, döntéseinek áttekintése.
3. Kapcsolat kiépítése és fenntartása más, e témakörökben tevékenykedő csoportokkal, kölcsönös információcsere.
 - 3.1. Adatbázisrendszerekkel szemben támasztott követelmények tanulmányozása.
 - 3.2. Reprezentatív rendszerek és megközelítések tanulmányozása.

4. Adatbázis rendszerek strukturájának és komponenseinek definiálása.

5. Iterálni az előbbieken kifejtett lépések folyamatával a következő esetleges kimenetek valamelyikének eléréséhez:

5.1. Adatbázis rendszerek olyan részének azonosítása, ami alkalmas szabványosítás kezdeményezéséhez, vagy vezérvonal kialakításához. Erről SPARC/90 riport készítése, melyben igazolni kell ezt a helyzetet.

5.2. Olyan részek azonosítása az általános modellekben, amelyek jelenleg még nem érettek szabványosításra. Ehhez is SPARC/90 riportot kell készíteni ezen helyzet igazolására.

5.3. Olyan komponensek azonosítása, amelyekre van bizonyos racionális alap további akciókhoz (követelmény halmaz vagy probléma definíció), de az nem elégséges szabványosítási, vezérvonal kidolgozási elhatározás meghozatalához. Itt további tanulmányozást, kutatást kell ajánlani egyértelmű döntés meghozhatósága céljából.

1.3 Az adatbázis rendszer dinamikus működése általánosan

Az általános rendszer diagram (1.1. ábra) jobb alsó részén lévő "alkalmazási programrendszer", valamint a 7 és 12. interface-k alkotják azokat a nem-adatbázis tevékenységeket, amelyek a teljes információs rendszerekben felmerülnek és az alkalmazói programok előkészítésének, végrehajtásának tevékenységét jelentik. Ez a rész változatos alrendszerek együttesét képviseli, amelyek mind a 12-es felületen keresztül csatlakoznak, különbözve azon nyelv természete szerint, amit a programozó használ az ember/gép interface-n keresztül történő kommunikációban. Ez lehet COBOL, PL/1, ALGOL, speciális nyelvek, mint riport generátor, lekérdező

nyelv vagy felfrissítési akciót specifikáló nyelv, illetve valamely potenciálisan új típusú procedurális vagy probléma orientált nyelv. A megjegyzésre érdemes dolog ezzel kapcsolatban az, hogy az alkalmazói programokban minden adatleírás (séma információ) a 12-es interface felületen megy át az adatbázisból. (Ez egyébként a "két sémás" CODASYL javaslatban is így van, és a SPARC javaslat elvileg nem különbözik tőle e ponton.)

A rendszer diagram bal alsó részén a "rendszer programozó" és a "rendszer programozói alrendszer", valamint a 16 és 18 interface-ek nyújtják azt a lehetőséget, amit a rendszer programozó használhat ki, ha át akarja lépni az adatbázis rendszerhez való hozzáférés szokásos, logikai szintű módját. A rutin szerű rendszer karbantartás és módosítás ezen az alrendszeren keresztül történik, amitől kivételesen eltérés is lehet egyes implementációknál. Általában rendelkezésre áll az alkalmazói programozóknak is az installációs opció, ami megengedi, hogy ezeken a felületeken keresztül fordulhassanak a rendszerhez, de ez potenciálisan nagy veszélyeket rejt magában. (Szintén megjegyzendő, hogy ebben a rendszerrészben sem nyújt többet a SPARC javaslat, mint a CODASYL DBTG javaslat.)

A rendszer diagram felső részén lévő, három sémát tartalmazó rész az ami, a "gép által látott" és a "programozó által látott" kétsémás CODASYL javaslattal szemben új a SPARC modellben (és újak ennek természetes következményei). Az Internális és Externális sémák elődjei tehát megtalálhatók az előbbi adatbázis rendszer modellekben is, bár kevésbé világos terminológiával. A Konceptcionális séma is jelen volt, de nem használták explicite. Ez a vállalat szemlélete arról a strukturáról, aminek modellezését megkísérli az adatbázisban. Ezt a szemléletet hívjuk be (eddig is ez történt a "megbizó" behívásával), ha vita van a programozó és a feladat felelős között, hogy mit értsünk pontosan a program specifikációja alatt. A SPARC munkacsoport meggyőződés-

se, hogy az adatbázis környezetben explicitté kell tenni a koncepcionális sémát, másrészt számítógéppel közvetlenül olvashatóvá, elérhetővé kell tenni az ABKR számára. Nyilvánvaló szükségszerűség, hogy az internális és az externális sémák konzisztensek legyenek a koncepcionális sémában megjelenő szemlélettel.

Ez a konzisztencia elfogadható biztonsággal fenntartható és igazolható, ha a koncepcionális séma számítógéppel feldolgozható formájú. A SPARC munkabizottság fontos tevékenységének tartja, hogy analizálja a koncepcionális séma természetét, vizsgálja annak explicitté tétele, formális leírása szempontjából szóbaeső formalizmusokat. Ettől függetlenül (használható formalizmus hiányában is) vizsgálható, hogy mit jelent a koncepcionális séma jelenléte az ABKR dinamikus működése szempontjából az 1.1. ábra alapján.

A vállalati adminisztrátor definiálja a koncepcionális sémát és a lehetséges vagy célszerű mértékig érvényesíti azt. A séma bizonyos részeinek konzisztenciája ellenőrizhető mechanikusan is, de mivel egy vállalat összes érdekes aspektusainak formális modelljéről van szó, az ellenőrzési feladat nagyon komplex lehet és a logikai inkomplettség, ellentmondásosság, nem egyértelműség felderítése is kívánatos. A koncepcionális séma tartalmazza - sok egyéb információ mellett - minden átfogni kívánt entitás definícióját azon tulajdonságok meghatározása révén, amiket a sémához relevánsnak tartunk. A definiált entitások közötti kapcsolatok szintén magyarázva lesznek az adatok megengedhető értékeivel együtt és egyéb rájuk vonatkozó korlátozásokkal. Közvetlenül modellezhetjük az adatahoz való hozzáférés szabályait azon személyek definiálásával, akik bizonyos tevékenység során az ABKR-t használják. Vagyis biztonsági, felületi rendet adhatunk a koncepcionális séma szintjén.

Jól tudott, hogy az eddigi ABKR implementációkban lényeges problémák vannak a biztonság értelmezésével, ezért a SPARC reportban a rendszerműködés irányításában központi szerepet játszó koncepcionális sémába építik be a biztonság kérdéskörét.

Az adatbázis adminisztrátor az internális séma definiálásáért felel, amely magába foglalja a választott tárolási stratégia absztrakt leírását, ami adott esetben realizálva lesz az ABKR-ben. Függetlenül a választott hierarchikus, hálózat, relációs, invertált vagy egyéb tárolási módtól, az adat mindig az internális sémában lesz tárolva. Az internális séma hordozza az adat "belső szintaxisát". (pl. numerikus értékek radixja, használt kódolási sémák, mértékegységek, stb.) Az adat reprezentációk közötti hozzáférési utak és relációs kapcsolatok formája is definiálásra kerül. Mindezen információknak konzisztensnek kell lennie és deriválhatónak a koncepcionális sémából. Ennek érdekében az adatbázis adminisztrátor minden részletében olvashatja a koncepcionális sémát. Az internális séma processzora (1.1 ábra) mechanikusan ellenőrzi a konzisztenciát. A konzisztencia követelmények által támasztott korlátoktól eltekintve az adatbázis adminisztrátor szabadon változtathatja és választhatja meg az internális sémát az adatbázis rendszer működésének optimalizálása céljából. Megfelelő interpreterek használatával lehetséges az internális struktúra dinamikus átszervezése is a normál működés fenntartása mellett. Tekintve az adatbázisok nagy adattömegét ez lényeges követelmény lehet, amit az implementációk tapasztalata szerint csak a felhasználó szemléletének és az adat számítógép általi szemléletének szétválasztása garantál, feltéve, hogy a koncepcionális séma megengedi azt.

Az alkalmazói adminisztrátor(ok) definiálja(k) az exterrális sémákat (ez a DBTG alsémája), ami az alkalmazói programozó (vagy a végfelhasználó) adatazsemlélete. Minden nagyobb alkalmazói területnek saját adminisztrátora van, aki annak

terület releváns sémáit definiálja, és amelyek a teljes adatbázis részeként jelennek meg. Ezeket látja az alkalmazói programozó és külön definiálás teszi lehetővé az adatnév különbözőségek feloldását. Az adatnév feloldás bonyolultságát, szimbolikus nevek kapcsolásának módját nem vizsgálják a javaslatban. Lehetőséget ad a javaslat arra, hogy ez fordításkor, rész programok behívásakor vagy végrehajtásakor történhessen meg, de ettől függetlenül a 7, 12 és 31 interface-ken keresztül történik meg az adatnevek kötése. Az ebben részt vevő sémák közötti alternatív választási lehetőség az 5-ös interface-nél történhet meg.

Az externális sémáknak a koncepcionális sémával való konzisztenciájára ugyanazok a megjegyzések érvényesek, mint amiket az internális séma tárgyalásánál vázoltunk. További lehetőség, hogy valamely externális séma valódi alhalmaza egy másik externális sémának és így a konzisztencia tranzitivitásának hipotézise mellett az externális séma processzor érvényesíthet valamely externális sémát egy sokkal átfogóbb externális sémával szemben, ami viszont konzisztens (előbbi vizsgálat eredményeként) a koncepcionális sémával.

Miután a megfelelő sémákat definiáltuk a rendszer működésének dinamikája egyszerűen felvázolható. Az alkalmazói programozó (riportok specifikálója, lekérdezéseket végző operátor, stb.) szokásos módon végzi munkáját a megadott externális sémával és mind explicit mind implicit módon az adatdeklarációval összhangban procedurális utasításokat ad a 7-10 interface felületen. Így aktivizálhat fordítást és más releváns adakezelő/kiszolgáló folyamatokat az alkalmazói program alrendszeren keresztül. Végrehajtáskor az adatok iránti kérelmek a 12-es felületen keresztül haladva jutnak koncepcionális/externális transzformátorhoz, ami megvalósítja a leképezést az externális és a koncepcionális adatleírások között. Ez a leírás jut át a 31-es felületen a koncepcionális/internális transzformátorhoz,

ami a koncepcionális és az internális leírás közötti leképezést határozza meg.

Az internális és a koncepcionális sémák általában statikusak, így a leképezés bonyolultságától függően, valamint az implementálás természetétől függően a két transzformátor összevonható egyetlen összetett leképzési függvény meghatározásával. Ez nem zavarja azt a képet, hogy a magasabb szintű adat függetlenség biztosítása céljából gyakran szükséges lehet, hogy a folyamatot kétfokozatunak írjuk elő.

Az adatkeresés az előbbi transzformáció (kétlépcsős vagy összevont) után a 30-as felületen az internális/tároló transzformátorhoz jut. Az internális séma a tárolót lineáris, több kezdőponttal bíró cimtérként ismeri és ezt az absztrakt modellt kell a fizikai tároló közeg hardware jellemzőibe átképezni (sávok, cilinderek, lemezek, stb.). Ez a "piszkos" tároló leírás ezután a 21-es felületen keresztül jut át a számítógép hardware szintjére, ahol valószínűleg további transzformációkon megy keresztül, amíg az aktuális adatot megkapjuk és az előbbi folyamatot visszafordítjuk. Az előbbi rövid leírás valamely logikai adategység megkapását tartalmazta, de a felhasználói program által előkészített adat betöltése is hasonlóan történhet.

Az előbbi működési sémában a "deadlock" (folyamatok kölcsönös blokkolása) feloldás, a biztonsági, integritási és egyéb ABKR kérdések helye is megvan, de azokkal most nem foglalkoztunk. Ezek nagyrészt különálló, zárt problémaköröket alkotnak a három szintű adatmodellben és szemléletben, amelyek lényegileg azonosak azzal, amit a kétsémás DBTG javaslat tartalmaz. Bizonyos esetekben - pl. a biztonsági védelem mechanizmusa - könnyebb megoldást nyújt a 3 sémás megközelítés.

II. ELVEK, MODELLEK, FOGALMAK, DEFINÍCIÓK

Egy elv megértése magába foglalja az elv osztályozását, összevetését más hasonló elvekkel és a köztük levő különbségek meghatározását. Az adatbázis és információs rendszerek elvének megértéséhez szükséges bizonyos definíciók egyértelmű használata. Az itt szereplő fogalmak értelmezése helyenként precízebb vagy korlátozottabb, mint az illető fogalom szokásos értelme. A definíciók célja annak rögzítése, hogy mikor és hogyan használjuk ezeket a kiterjesztéseket. A következőkben leírjuk röviden azt a környezetet, ahol ezeket a fogalmakat használjuk és megadjuk a fogalom definícióját, szerepének, használatának körülírásával. Ezzel keretet adunk a teljes adatbázis rendszer működéséhez is. A rendszer működéséhez különösen fontos szempontokra (adatfüggetlenség, felhasználók típusai) és a funkciók ellátásában központi szerepű rendszer komponensekre (adatszótár/tartalom jegyzék) külön is kitérünk.

2.1. Az érdeklődés birodalmai és az adatbázis

Az információ filozófiájában három birodalmat szokás megkülönböztetni:

- a) a valós világ, ahogyan az fizikailag létezik.
- b) fogalmak a valós világról, ahogyan az emberek fejében léteznek.
- c) szimbólumok papíron vagy másfajta tároló közegen az előbbi fogalmak reprezentálására.

Mindhárom birodalomban az információnak vannak sajátos tulajdonságai, amelyek erősen különböznek egymástól.

Az információs birodalmakon túl az adat feldolgozásban (ami a valós világbeli fogalmak szimbólikus reprezentációinak manipulálásával foglalkozik) van néhány további adatbirodalom. Ezek közül három játszik központi szerepet a SPARC adatmodellben:

- a) externális adatbirodalom jelenti a valós világ egyszerűsített modelljét, amint azt egy vagy több alkalmazás látja, illetve megkívánja.
- b) koncepcionális adatbirodalom jelenti a valós világ korlátozott modelljét, amit az összes szóba jövő alkalmazáshoz tartunk fenn központi referenciaként.
- c) internális adatbirodalom jelenti a valós világ azon korlátozott modelljét, amit a számítógépes tárolókban tartunk fenn adatok formájában.

Hasonlóan az információs birodalmakhoz, a három adatbirodalombeli adatok is erősen különböznek egymástól.

Szükséges, hogy az adatbázis rendszerben megjelenő objektumok birodalmát megkülönböztessük, hogy a hivatkozás hozzájuk helyes legyen és a megfelelő elvet alkalmazhassuk. A jelen leírásban részben kvalifikátorokkal, részben csak a birodalomban használt névvel adjuk meg az objektumok hovatartozását.

Az egyes birodalmak adatmodellből és sémából állnak, ami leírja a vonatkozó modellt. A valós világ egy objektuma az entitás. A modellekben a vállalathoz tartozó entitások kollektívái és a rájuk vonatkozó tények vannak reprezentálva. Az egyes modelleket azért hozzuk létre, hogy olyan teljességgel és torzítás nélkül reprezentálják a vonatkozó entitások számunkra érdekes tényeit, amennyire az igazolva van az adott kontextusban. A modellek precizitása "trade-off" tárgya az alkalmazás követelményei (előnyök) és a gazdasági igazolhatóság (költség, ráfordítás) között. A modellekben minden objektum osztályozva van és minden objektum tipushoz leíró információ (deskriptorok) tartozik. Az egyes adatbirodalmak sémája a modellekre vonatkozó deskriptorok gyűjteményét jelenti.

Valamennyi információs és adatbirodalom egyetlen vállalat (hipotetikus vagy valóságos) számára fontosnak tekintett tényeket tartalmaz. A definiált tények reprezentációjaként előálló adatkollekció az adatbázis. Ez jelent egyrésről egy hozzágondolt (nem-diszjunkt) koncepcionális rekordhalmaz kollekciót, ami a vállalat koncepcionális modellje. Ezen túl az adatbázis (diszjunkt) internális rekord-halmaz kollekció, ami a fizikailag tárolt adatot tartalmazza és ez egyben az internális modell. Egy vállalat több adatbázissal is rendelkezhet, amikor az egyes adatbázisok a vállalat egyes (névlegesen diszjunkt) részeire vonatkozó adatot tartalmaznak.

Minden adatbázishoz tartozik egy állandó fejlődésben lévő internális séma, ami az internális modellt írja le, egy állandó fejlődésben lévő koncepcionális séma, ami a koncepcionális modellt írja le, továbbá annyi externális séma, amennyit a különböző externális modellek megkívánnak.

2.2 Általános használatu fogalmak

A három sémás ABKR modell egyes birodalmaiban sajátos fogalmak is előfordulnak, de számos fogalom közös értelmezésű mindenütt. Vannak generikus fogalmak, amelyek árnyalatnyi eltérést mutatnak attól függően, hogy melyik sémában jelennek meg. A következőkben ezeket a fogalom csoportokat részletezzük.

2.2.1 Objektumok fogalmai valamennyi adatbirodalomban

- Objektum: Az objektum a valós világban illetve annak valamelyik modelljében megjelenő egyed (entitás, "valami").

Különböző dolgok, entitások és azok reprezentációja lehet vagy egyéb olyan információ objektum (annak reprezentációja,) ami nem feltétlenül entitás reprezentáció.

Az objektum külön osztály vagy típus lehet, aminek leíró jellemzői (deskriptorai) vannak, amik megfelelnek az illető osztálynak vagy típusnak (meghatározzák vagy megkülönböztetik másoktól). Előfordulásai vagy esetei vannak, amik követik a deskriptorokkal leírt sajátságokat. Általában (és ebben az adatmodellben is) objektum osztály és típus, objektum deskriptor és előfordulás között csak akkor teszünk különbséget, ha valamely funckió, tevékenység kifejtéséhez ez szükséges.

- Reprezentáció: Entitások vagy más objektumok képét, szimbólumát, egységesített jelét (zsetonját) nevezzük reprezentációnak. (pl. város térképpel reprezentálható, egy dolog tulajdonosságát adatmező aktuális értéke jeleníti meg stb.). Magába foglalja a egységesített jel megvizsgálását és megtestesülését valamilyen médiumon, azaz létezik a jel leképzése (mentálisan vagy materiálisan) a megtestesült megjelenítésbe. A leképzés lehet nagyon bonyolult és indirekt, vagy egyszerű, közvetlen a teljes kongruensségig terjedő. Az egységesített jel és annak megtestesülése között megkívánt kongruenciát a leképzés megvalósításához rendelkezésre álló idő vagy mentális/materializálási leképzés bonyolultsága (annak megengedhető szintje) határozza meg.
- Attributum, szerep, és tartomány: Ezt a három fogalmat a koncepcionális sémában definiált objektumokkal kapcsolatosan magyarázhatjuk legkönnyebben, de alkalmazhatók a többi séma objektumaira is. Az externális séma azonban alakítható valamely egyedi nyelv vagy alkalmazás család szempontjai szerint, amikor a szerep és tartomány koncepciója takarva lehet a használt szimbolizmussal.

Egy attributum (koncepcionális mező) valamely entitás tulajdonságának reprezentációja. A szerep ennek a koncepcionális mezőnek azon funkciója, amit a koncepcionális rekordban játszik, hogy leírjon egy egyedet az entitás halmazban. A tartomány azon értékek teljessége, amelyből az illető koncepcionális mezőre érvényes értékek választhatók. Az aktív tartomány a jelenleg ismert tényeket reprezentálja létező entitásokról. Ugyanazon tartománybeli értékek mindig összehasonlíthatók, még ha reprezentációjuk eltérő is, viszont a különböző tartományok értékei akkor sem, ha reprezentálásuk összeesik.

Valamely attributum mindig szerephez tartozik és van tartománya. Szokásos gyakorlat, hogy az attributum megnevezését célzó azonosító tartalmazza mind a szerep megnevezését, mind a tartomány nevet ("fizeti osztályhoz"). Emberi okokból és formátumozási célszerűségből az a szimbólum-sor, amit riportok fejében vagy programban használunk, rövidebb és helyileg emlékeztető jellegű (mnemonikus) szinonima a rendszerbeli attributum névhez, ami a teljes szerep nevet és a tartomány nevet foglalja magába. Az attributum által betöltött funkció, szerep és tartomány az adatmezők érvényesítésénél és kapcsolatok létrehozásával lesz kihasználva.

Az attributum (metaszerep) szerep funkciója lehet azonosítás, történet, összegezés, irányítás, státusz jellemzés, leírás, struktúra képzés, stb. A szerep struktúra képzés, ha koncepcionális rekordok között létesít kapcsolatot (tükrözve a hozzájuk tartozó objektumok közötti kapcsolatot). Leírás természetű a szerep, ha viszonylag stabil információt képvisel. Státusz jellemzést lát el a szerep, ha olyan a vonatko-

zó információ, ami állandóan változik annak érdekében, hogy tükrözze a jelen állapotot (rendelkezésre álló áru mennyiség). Az attributum szerepe irányítás természetű, ha metaadatnak tekinthető (pl. iterációk száma többértékű attributumra vagy egy kapcsolatos rekordra). Összegzés a szerep, ha az attributum függvényt reprezentál (összeg, átlag, stb.) a megfelelő értékre egy vele kapcsolatos egyedi kollekcióból. Történeti szerepet játszik a koncepcionális mező, ha olyan információt képvisel, amit időben túlhaladtunk (készlet a múlt hét végén). Az attributum vagy azok kombinációja azonosítást végez, ha valamely egyed (egyértelmű) azonosításához szükséges információt hordozza.

Az előbbieken túl több szerepe is lehet valamely attributumnak, amelynek intuitive adható értelmezés. A legjelentősebb szerepek az előzőek közül az azonosítás és a struktúra képzés, de ezek sem kizárólagos szerepek, mivel a struktúra képző szerep egyúttal leíró jellegű is. A leírószerp akkor válik struktúra képzővé, ha koncepcionális adatot kezdünk fenntartani attributumokról, illetve olyan adat (szerepre vonatkozó koncepcionális adat), hogy megszüntetése után a struktúra szerep is megszűnik. Például egy személyre vonatkozóan az az információ, hogy ő programozó, leírás szerepű (lehet státusz is). Ha azonban szakképzettségre vonatkozó leírásokat tartunk fenn ahhoz, hogy az illető programozót kapcsolatba hozzuk jelenlegi munkájának leírásával, akkor a "szakképzettség" attributum struktúra képző és leírás szerepűvé válik. Ha az azonosító attributumok kombinációja, akkor ezen attributumok némelyike nyilvánvalóan leírás és/vagy struktúra képző szerepű egyidejűen.

A tartomány szót széleskörben használják az adatfeldolgozás szóhasználatában (pl. alkalmazási terület, ami felett azonos a használt eljárás; objektum együttes, ami közös definíciót fogad el egy szimbolumra, stb.). Itt "érték tartomány" értelemben használjuk és legfontosabbnak a tartomány elfogadhatóságának szabályait tekintjük. Ezek változnak bonyolultságban az egyszerűbbtől (pl. súly 1-99 között) az összetettebbekig (pl. megengedhető szimbolumok egy táblában vannak, maszkokkal vagy nyelvi utasításokkal leírt tartományok stb.). Amint említettük ugyanazon tartománybeli értékek összehasonlíthatók, még akkor is, ha reprezentációjuk eltérő a különböző objektumokra (pl. dátumok megjelenítése). Fordítva, bár a "KOVÁCS" érték a nevek és foglalkozások tartományában azonos reprezentációju lehet, mégsem összehasonlíthatók.

A tartomány jellemzők érvényesítésre történő felhasználása nyilvánvaló és jól ismert. A tartománybeli legális értékeknek bizonyos szabályokat kell kielégíteniük ahhoz, hogy valamely koncepcionális rekord attribútumához rendelhető legyenek (értéktartomány és forma összefüggések szokásosak). A kapcsolat (struktúra) képző szerep bonyolultabb és fontosabb az érvényesítésnél.

A kapcsolat alapvetően abból ered, hogy két vagy több objektumban valami közös, amit többféleképpen lehet reprezentálni (egyutas pointer lánc, faktorizált

értékek (hierarchiák), implicit értékek (szomszédosság)). Mindezen reprezentációk lokális optimalizálás eredményei valamely (normalizált) reprezentációban arra a tényre, hogy az érintett objektumok kötött értéket tartalmaznak. Az érték az egyik objektum vonatkozó mezőjében olyan szerepű, hogy kapcsolatot hozzon létre a másik objektummal, amiben az illető érték szintén egy mező. Gyakori, hogy magát a kapcsolatot a szerepnév jellemzi.

Az előbbiekből megfogalmazható a következő szabály: kapcsolatok csak olyan értékek reprezentációján keresztül létesíthetők, amelyek ugyanazon tartományból származnak, függetlenül attól, hogy szerepnevek közösek vagy sem; ha az értékek ugyanazon tartományból származnak, akkor azok kapcsolatot hoznak létre a szerepnevektől függetlenül (a kapcsolat lehet potenciális, fel nem ismert vagy fel nem használt). Ha az értékek nem alkotnak tartományt, akkor nem képezhetnek kapcsolatot, még akkor sem, ha a szimbólumok vagy a szerepnevek azonosak is. Más szavakkal a kapcsolatokat közárólag struktúra képző szerepű mezők hozzák létre értékkel, pointerekkel, implicit vagy rejtett mezővel reprezentálva, explicit/implicit megnevezett strukturális mezővel, stb. és a mezők csak akkor struktúra képző sajátosságuk, ha ugyanazon tartományból származnak.

Érték: Bármely modellben egy entitás vagy egyéb objektum tulajdonságainak előfordulása, illetve ezen előfordulás reprezentációja. Az érték lehet mennyiségi vagy minőségi tulajdonság megjelenése. Általában a sorsszámok a neveket, tőszámok mennyiségeket fejeznek ki. Aritmetikai műveletek sorsszámokon is végezhetők (sőt minden bitsorozattal kifejezett szimbólumon), de ennek eredménye szokatlan és váratlan reprezentációhoz vezethet.

Azonosító: Az azonosító olyan tulajdonság vagy azok kombinációja, amelyek értékei entitások vagy más objektumok egyezményes szimbólumaiként, illetve megnevezéseként szolgálnak. Az egyezményes szimbólum lehet általánosan használt név, érték, értékek kombinációja (amiket nem használunk általánosan megnevezésként), önkényesen hozzárendelt szimbólum (pl. rendszer azonosító szám, adatbázis kulcs), amit nem ismernek a hozzárendelési mechanizmus vagy rendszer határain kívül. Ha az illető entitás mindig megkülönböztethető a választott azonosítóval a többi entitástól, akkor az azonosító egyértelmű. Ugyanannak az entitásnak több azonosítója is lehet, amiket különböző alkalmazások használnak, de szokásos ezek közül egyet elsődlegesnek kijelölni. Valamely azonosító lehet csak egy alkalmazáson vagy szűkebben kijelölt objektum halmazon belül egyértelmű.

Az egyes modellekben az azonosító értékeket kulcsnak nevezzük. Ha az entitások egyértelműen azonosíthatók egy meghatározott azonosítóval, akkor a vonatkozó kulcsok egyértelműek lehetnek. Szokásos módszer, hogy az elsődleges kulcsokat úgy választjuk, hogy egyértelműek legyenek és ne kelljen őket változtatni az azonosított entitás, objektum élete során az egyes adatbirodalmakban (koncepcionális, externális, internális rekord azonosítók stabilisak legyenek).

- Osztály, típus: Ez a két fogalom gyakran szerepel az adatfeldolgozási szóhasználatban és nagyjából azonos lehetőséget adnak entitások és más objektumok osztályozására:
 - az osztály olyan objektumok kollektívja, amelyek hasonlóságát az adja, hogy azonos prototípussal rendelkeznek,
 - a típus jelenti a prototípust objektumok olyan kollektívjára, amelyek hasonlóak.

Objektumok valamely osztályának neve objektum kollektívok neve, amelyek hasonlóak, azaz olyan dolgok, amiknek azonos fajta tulajdonságai vannak (deskriptív tulajdonságaik azonosak). A hasonlóság kritériuma az, hogy önkényesen választott szabály halmazzal konformok (a szabály halmaz egyedeinek megkülönböztetését vagy azonosságát hangsúlyozhatja). A szabály halmaz definiálja az osztály minden tagját, valamint azt a prototípust, ami az illető osztály tipikus

tagja. A prototípus neve rendszerint ezen objektum típus neve. Valamely alkalmazás kapcsolatos lehet egy entitás osztállyal, amit egy entitás típus tipizál. Ez lehet egy externális rekord osztály, amit valamelyik externális rekordtípus tipizál. Így lehet osztályozni objektum kollekciókat akár típusuk azonosításával, akár osztályuk meghatározásával. Vagyis lehet egy objektumot asszociálni típusának leíró tulajdonságaival (deskriptoraival) vagy osztályának megnevezésével. (Az internális modell egyik objektumát jelentő forma terjedelem definíciója és ezen definíciónak az internális modellbeli definícióval szembeni kvalifikációja nyilvánvaló). Az "osztály" fogalmat nem használjuk matematikai értelemben.

2.2.2 Kapcsolatok és kollekciók

- Kapcsolat: a valós világban két vagy több entitás kollekció vagy entitás tulajdonság között létezhet kapcsolat. A kapcsolat magukat az objektumokat, a kapcsolat fajtáját és irányát foglalja magába. A valós világban létező kapcsolatok nagyon összetettek lehetnek. Hasonló vagy különböző objektumok között létezhetnek nem irányított, egyirányú, páronként irányított, tranzitív, nem-tranzitív, rekurzív, konverz, reciprok vagy az előbbiektől különböző egyéb kapcsolatok. A kapcsolatok időben változhatnak, fejlődhetnek, egyidejűen több kapcsolat létezhet. Az is gyakori, hogy két objektum osztály egyidejűen több mint egy kapcsolatot tart fenn egymás között.

A valós világ objektumai közötti kapcsolatok gazdagabbak és árnyaltabbak, mint amit modellezni jelenlegi eszközeinkkel képesek vagyunk. Ezen kapcsolatok reprezentációi erősen egyszerűsítettek és stilizáltak. Gyakran csak a kapcsolat ténye kerül deklarálásra, annak fajtája (szemantika, kontextus, jelen státusz, stb.) nem. A modellekben explicite megjelenő kapcsolatokról nem mindig derül ki, hogy azok valós világbeli kapcsolatok megfelelői, vagy félrevezető kapcsolatok valós világbeli megfelelő nélkül. Gyakori az is, hogy egy kapcsolat értelme és megjelenése beépül abba az algoritmusba, ami feldolgozza az illető objektumokat, ezért a modellben nem kerül explicit kifejezésre az illető kapcsolat.

- a) Hozzárendelt kapcsolatok: gyakran egy nyilvánvaló, de nem jelentős kapcsolatot kihangsúlyozunk objektumok halmazán annak érdekében, hogy valamely hozzáférési módszerhez szervezzük őket. Például internális rekordok rendezhetők alfabetikusan az egyes rekordok kényelmes elhelyezéséhez. Ezt a sorrendet az alkalmazások ismerik és kihasználják a teljes rekordhalmaz végig pásztázásánál vagy a működési (fizikai) sajátságok javításához, miközben páronként illesztünk externális rekordokat két vagy több externális rekord halmazban. Ugyanakkor ez a ráakott kapcsolat, ami tükrözi az adattároló szervezést nem hordoz információt, mivel az objektumok kialakuló sorrendje (ahogy tárolásra kerülnek) nincs analógiában egyetlen valós világbeli sorrenddel sem. Ezzel a nyilvánvalóan létrejövő kapcsolattal szemben a "fizetés" szerinti sorrend valóságos, információt hordozó kapcsolat. Ez az oka annak, hogy az ABKR ilyen jellegű kapcsolatokhoz való hozzáférést védi

és irányítja a koncepcionális sémában megadott deklarációkkal összhangban, míg számos rárakott kapcsolatot nem véd.

b) Félrevezető kapcsolatok :

Keletkezhetnek objektumok között hibás, vagy félrevezető kapcsolatok is, amelyek nyilvánvalóan nem érvényesek vagy értelmetlenek, mivel álokoskodás, véletlen szomszédosság vagy hasonlóság szülte őket. A megfigyelt jelenség helytelen interpretálásából származnak ha a véletlen vagy relative gyakori koincidenciából túl hamar extrapolálunk. A látható kapcsolatok ugyanis ténylegesek lehetnek bizonyos esetekben, de nem másokban. Ezért szokásos, hogy bármely kapcsolat bevezetése az információs rendszerbe kivételes jel (nyelvi forma) használatával történik.

A modellezés vagy eszközeink korlátai miatt szükséges tulegyszerűsítés vezethet szintén félrevezető kapcsolatokhoz. Az egysikuan, alkalmazás orientáltan szemlélt objektum halmazokból származtatott kapcsolatok elfedhetik más alkalmazások szempontjából fontos (létező) és eltérő jellegű kapcsolatokat.

c) Nemdefiniált kapcsolatok:-

Valós világban létező kapcsolatok ismeretlenek maradhatnak az információs rendszer számára szándékos tervezési megfontolásból, a kapcsolat irreleváns volta miatt, vagy tudatlanságból. Entitásokra (pl. személyekre) vonatkozó tények hordozhatnak olyan kapcsolato-

kat, amelyeket "privacy" okokból nem tarthat fenn explicite az információs rendszer, mivel a felsőbb vezetés dönthet így. Ugyanakkor lehetséges, hogy statisztikus, korrelációs kapcsolatokat megengedünk explicit formában, miközben védjük az egyén személyiségét. A rendszerek fejlődése a kezdetben irrelevánsnak ítélt (de valójában fontos és létező) kapcsolatokat és a tudatlanságból elhanyagolt kapcsolatokat definiálttá változtatja, de az előbbi csoport mindvégig nem definiált marad törvényességi okokból.

Az előbbieken entitások, objektumok közötti kapcsolatok infinitive mnemonikus neveket hordoznak. A valós világbeli kapcsolatok fajtáinak gazdagsága egyszerűsítetten jelenik meg a különböző adatbirodalmakban. Az externális és a koncepcionális modellekben a kapcsolatok sokkal pontosabb definícióval jelennek meg és sokkal stilizáltabb nevük van, mint az internális modellben (itt bizonyos esetekben az adatstruktúra technológiákban bizonyos kapcsolatokat nem is kell megnevezni explicit módon).

Az externális és a koncepcionális sémákban egy kapcsolat összefüggést jelöl két vagy több objektum között, ami kiválasztási utat specifikál minden azonosítható objektumra. A legegyszerűbb és a valósághoz közeli példák a koncepcionális séma objektumai köréből hozhatók, de ugyanezek a megállapítások igazak az externális modellben is. A kapcsolat két vagy több koncepcionális mező vagy csoport összefüggés lehet ugyanazon koncepcionális

rekordban, két vagy több koncepcionális rekord közötti kapcsolat ugyanazon vagy eltérő koncepcionális rekord halmazokban. Nyilvánvaló, hogy ezek az objektumok azonosíthatók azon kapcsolat révén, amiben részt vesznek, továbbá az objektumok bármely más tulajdonságával is.

Mindegyik modellben a kapcsolat az illető modell két vagy több objektuma közötti összefüggés. Az egyes modellek objektumai és kapcsolatai leképezhetők a másik modell vonatkozó objektumaiba, illetve kapcsolataiba. A kapcsolatokat reprezentálhatjuk "juxtapozícióval", alárendeltséggel objektumok között; 1:több vagy több: több irányított kollektcióként, ami azonos típusu vagy különböző típusu rekordok előfordulását köti össze. Kifejezhető a kapcsolat "metszet" rekord formájában is, amikor a rekord mezői különböző koncepcionális rekordok mezőiből adódnak, amelyeknek közös koncepcionális mezeje van. Az egyik legismertebb kapcsolat képzési mód (lásd relációs adatmodell) közösen használt tartomány azonos értékein keresztül történik. Mivel a változatos megjelenítési mód nem része a kapcsolat fogalomnak, a kapcsolat reprezentálásának eszközeit nem tekintjük valódi kapcsolatnak.

A kapcsolat információ hordozó karaktere az externális(koncepcionális) modellben szembeállítható az internális modellbeli hozzáférési uttal és annak szervezésével.

- Asszociáció: Zéró vagy több kapcsolt objektum kollektciója nem-strukturált vagy nem-irányított kapcsolat révén. Például alkatrészek

együttese szállító szalagon (bár lehetséges strukturát rárakni erre a kollekcióra, de az asszociáció az ilyen strukturáktól mentes kollekciókat jelenti). Az asszociáció objektumai tételek, asszociációk vagy strukturák lehetnek. (Az asszociáció matematikai neve halmaz, ha az objektumok egyértelműek.)

- Struktura: Irányított kapcsolatokkal (szintezett, kölcsönös) összekötött zéró vagy több objektum kollekciója. Az alkotórészek sorrendezetten (rendezett) és/vagy hierarchikusan (szintezett) kapcsolva képzik a kollekciót. A stratifikáció lehet lineáris, hierarchikus vagy hálózatszerű, stb. A strukturát alkotó komponensek elemi tételek, asszociációk vagy strukturák lehetnek. A struktura lehet valamely halmaz tagja, másrészt halmaz lehet struktura alkotó komponense, továbbá egy struktura komponensét jelentő objektumok halmazt képezhetnek (struktura komponens mi-voltukon túl). A struktura komponensei nem képeznek matematikai értelemben halmazt.
- Rész halmaz: Ez olyan *matematikai értelemben vett halmaz*, amelynek elemei egy *befoglaló halmaz* elemei: Ebben a reportban gyakran nem matematikai értelemben, hanem szabadabban használjuk *nagyobb egység darabjának, töredékének* jelölésére.
- Szervezés: Internális adatok és rendszer orientált metaadat (kötet címkék, internális rekord-halmaz címkék, adathalmaz címkék, indexek, pointerok, stb.) fizikai elrendezése az internális modellben, ami hozzáférési utat biztosít az internális adat minden azonosítható (többféle módon is) előfordulásához. A hozzáférési ut, mint eszköz egy in-

ternális rekord megkereséséhez (helye szerint) az internális adat tároló szervezetben összehozható a kiválasztási uttal, mint eszközzel, amivel externális rekordok választhatók ki (jellemzők szerint) externális rekord halmozokból vagy azokkal összefüggő struktúrákból.

2.2.3 Generikus objektumok

Az adatfeldolgozási gyakorlatban kialakult néhány objektum, amelyek központi szerepet játszanak az adatbirodalmak objektumai között vagy azok képzésénél. Ezek mindhárom modellben megjelennek, de szignifikáns differenciáik ellenére (ezeket külön tárgyaljuk) közös definiálásuk is lehetséges általános fogalomként. Bár van generikus definíció, nincsen generikus objektum, azaz nincsenek "rekordok", hanem koncepcionális, externális vagy internális rekordok léteznek és mindig valamelyik jelzővel együtt használjuk a generikus fogalmakat. Ezekre az objektumokra nézve kialakultak az általános műveletek is (tárolás, címzés, visszakeresés, módosítás, törlés, stb.) amiket szabadon, az eddigi gyakorlat során kialakult intuitív jelentéssel használunk a következőkben. Hasonlóképpen a "logikai" és "fizikai" kvalifikátorok is a szabad értelmezés szerint kerülnek használatra.

- Mező: A modellekben előforduló legkisebb névvel ellátott objektum, ami érték hordozója, vagyis ez kerül módosításra ha egy érték, tény megváltozik. Reprezentálhat algebrai vagy logikai (Boole) mennyiséget, illetve egyéb szimbólikusan kifejezett minőséget. A mező atomi értékkel bír, ha tovább osztva ezen részekhez már nem rendelhető értelem. Ha az érték nem atomi természetű, akkor csoport-

- értéknek tekintjük. A mezőknek nevük, leíró tulajdonságaik, előfordulási sokaságuk van. A mezőt adatfeldolgozási (nem matematikai) értelemben használjuk.

- Csoport: A csoport zéró vagy több mező és/vagy csoport megnevezett strukturája vagy asszociációja. A mezők és csoportok azonos, vagy különböző típusúak lehetnek a csoportban. Bármely csoportba tartozó mező vagy csoport előfordulása komplett az illető csoport előfordulásában. A mezőket általában két okból vonjuk össze csoportba: közösen címezhető mezők asszociációját akarjuk képezni, vagy több értékű mezőkre indexelhető vektorokat, tömböket akarunk létrehozni. A csoportnak neve, leíró tulajdonságai, előfordulási sokasága van. A csoportot adatfeldolgozási (nem matematikai értelemben) használjuk.

- Rekord: A rekord zéró vagy több mező és/vagy csoport névvel ellátott strukturája vagy kollekciója. A rekordhoz tartozó mezők és csoportok egy vagy több típusúak lehetnek. A rekordokhoz tartozó mezők vagy csoportok előfordulása teljes az illető rekord előfordulásaiban. Ez az az objektum, ami logikailag vagy fizikailag tárolásra, törlésre, visszakeresésre, stb. kerül az egyes adatfeldolgozási alkalmazásoknál. Rögzített vagy változó számú elemből tevődhet össze. A rekordnak megnevezése (neve), leíró tulajdonságai (deskriptorok) és előfordulási sokasága van.

- Plex: A plex zéró vagy több rekord és/vagy plex névvel ellátott strukturája vagy asszociációja. A plexhez tartozó rekordok és plexek előfordulása teljes az illető plex előfordulásaiban. A plexek különböző konstrukciók lehetnek (pl. adatstruktúra halmaz, IMS adatbázis rekord, stb.) Képzésük motivációja az, hogy rekordokat olyan strukturákba gyűjtsünk, hogy címezhető/indexelhető vektort vagy tömböt kapjunk egy objektum előfordulásaira, amelyekre bizonyos közös-

ség lett difinálva, vagy amelyekhez ez asszociálható (pl. egy szerelvény különböző komponensei). A plex így alkalmazás orientált referencia mechanizmus, ami rögzített, vagy változó számú komponensből állhat. A plexnek neve, leíró tulajdonságai és előfordulási sokasága van.

- Rekord halmaz: Zéró vagy több rekord és/vagy plex névvel ellátott struktúrája, vagy asszociációja. Alkotó elemei azonos, vagy különböző típusúak lehetnek. A bel tartozó rekordok és plexek előfordulásai teljesek a vonatkozó rekord halmazokban. Ez az az objektum, amely megnyitható/lezárható és egyben a legnagyobb objektum amelynek neve, leíró tulajdonságai és egy vagy több előfordulás sokasága van (generációk, verziók, stb.).

2.3 A valós világ

Mindhárom adatbirodalom objektumai és azok reprezentációi a valós világ objektumainak közvetlen vagy közvetett megjelenései, vagy azokhoz kapcsolódnak. Az előzőekben vázolt modellalkotási folyamat hozza létre azokat az ideákat az ember fejében, amiket tükröztet a különböző szimbolikus modellekben. A következő fogalmak a valós világ objektumaihoz kapcsolódnak és a többi modell fogalmainak kiindulópontjaként szolgálnak.

- Entitás: A vállalat számára érdekes valóságos vagy absztrakt dolog, személy, hely, esemény, annak valamely része, ami külön választhatóan definiálható, megnevezhető, aminek révén ez a rész entitássá válik. Az ezen részbe foglalt tulajdonságok az újonnan kreált entitás tulajdonságai lesznek. Entitások kollekciói (asszociáció vagy struktúra) külön definiálhatók, névvel láthatók el és ennek révén entitássá válhatnak. Valamely "hozzátartozó" entitás tulajdonságai a befoglaló entitás tulajdonságai is egyuttal. Így az entitás típusok száma

és a különálló entitások száma lényegileg nem korlátos abban a kontextusban, ahol felhasználásra (definiálásra) kerülnek.

Különleges entitások fajtáinak nevezzük azokat, amelyek a rendszer és nem a vállalat számára érdekesek elsősorban. Ezeket rendszer objektumoknak nevezzük és a három séma valamelyikében kerülnek definiálásra (pl. koncepcionális rekord, vagy internális mező). Ezek az entitások objektumok, amelyek a vállalat adatbázisát alkotják. Az adatbázis számára nem ezen rendszer objektumok által reprezentált tények, hanem ezen reprezentációk érdekesek (maguk az objektumok). Például nem a megmunkáló sorszám, hanem a lyukasztott adatkártya (ami arra vonatkozó tényeket tartalmaz) és aminek jellemzői érdekesek (koncepcionális rekordok, internális rekordok, stb.). Ezen objektumok tulajdonságai magukba foglalja a méret (hossz) jellemzőket, bekódolást (kép-jellemzők), érték-halmaz definíciókat, stb.

- Tulajdonság: A tulajdonság entitás jellemzőt jelent (pl. személy neve, munkakör, stb.). Az entitások leírásánál, azonosításánál, jellemzésénél fontos szerepet játszik, továbbá kapcsolatot hozhat létre entitások között. A tulajdonságoknak tartományuk van, ami meghatározza a lehetséges értékeket az illető entitásra. Ugyanazon tulajdonság lehet több mint egy entitás típus jellemzője is. Például személyek, gyerekek, állatok valamennyien rendelkeznek névvel, sullyal vagy egyértelmű azonosító számmal. Nincs kizáró ok, hogy egy tulajdonság megengedett értéke személyre és más entitásra ne lehessen ugyanazon tartományu.
- Tény: Az entitásokra vonatkozó állítások tények formáját öltik megadva, hogy azon entitás valamely tulajdonságánál adott értéke van. Valamely tény állítás lehet igaz vagy hamis, de ez jelenti az illető entitásra vonatkozó ismeretet.

Ugyanazon tény vonatkozhat több mint egy entitásra is, például egy adott pénzösszeg kifizetés lehet egy személy felé, ugyanakkor költség összetevő egy berendezés vagy költségviselő osztály felé.

- Entitás halmaz: Az entitás halmaz egymással valamilyen módon kapcsolt entitások kollekciója, azaz egy vagy több tipusu entitás szerepel bennük, amiknek közös részük, fogalmi kapcsolatuk van és ez lényeges a vállalat számára. Egy entitás több mint egy entitás halmaz tagja is lehet egyidejűen.
- Vállalat: Személyek, események, irányítási stratégiák, folyamatok, információ és adatok kollekciója, ami logikailag strukturált és fizikailag úgy van szervezve, hogy bizonyos célok eléréséhez alkalmas legyen. Entitás halmazok kollekciója, amik logikai strukturában és a fizikai szervezetben jelennek meg. A vállalat jelenti azt a definiált és létező valóságos környezetet, amelyben az információs rendszer és az adatbázis rendszer működni fog.

2.4 Koncepcionális modell

A koncepcionális modell fogalmi állnak legközelebb a valós világ fogalmihoz és objektumaihoz, de ezen modell szerepe miatt és megjelenési formája következtében (szimbolikus modell, szemben a valós világ mentális modelljével) számos egyedi sajátossággal rendelkezik. A következőkben leírandó fogalmaknál és objektumoknál helyenként utalunk ezekre a közvetlen kapcsolatokra.

A koncepcionális modell olyan objektumok kollekciója, amelyek a vállalat fontosnak ítélt entitásait reprezentálják. Azt az objektumot, ami a koncepcionális modellben az entitást reprezentálja koncepcionális rekordnak nevezzük. Egy adatbázisban valamely entitáshoz tipikusan több externális rekord tartozik, de csak egy koncepcio-

nális rekord. (Ugyanez vonatkozik az externális modell és a koncepcionális modell esetére is).

A koncepcionális modell objektumait általában nem kell fizikailag megtestesíteni ("materializálni"). De az általuk játszott szerep jobban megérthető, ha úgy tekintjük, hogy az alkalmazás által látott objektum összeköttetésben van egy koncepcionális modellbeli objektummal és a koncepcionális modell objektumai leképződnek az internális modell objektumaiba. Ennek az itányított leképzésnek az a motivációja, hogy a koncepcionális adattal átfogó modellt hozzunk létre, biztosítsuk az adatok használata feletti irányítást, a közös és megosztott használatot, garantáljuk a programokba és az adatok létrehozásába investált erőfeszítések, ráfordítások védelmét. Az általános adatbázis rendszer az absztrakt koncepcionális modellt úgy tekinti, úgy hivatkozik rá, mintha a koncepcionális objektumok együttese valószínűségeként létezne.

- Koncepcionális mező (attributum): A koncepcionális adatobjektumok között legkisebb névvel ellátott egység, ami egy entitásra vonatkozó tényt, elvet, fogalmat reprezentál. Definícióját szerepének és tartományának meghatározására adja meg és szervesen hozzátartozik, hogy algebrai, logikai vagy egyéb szimbolikus érték tartozik hozzá. Konkrét megjelenítésre nem kerül a koncepcionális mező, ezért nincs formátuma vagy alakja (képe) vagyis reprezentáció forma (bitkép, karakterkép) nincs deklarálva értékére. Ugyiszintén nem rendelkezünk helyére vonatkozóan a koncepcionális exportokban vagy a koncepcionális rekordokban.

A koncepcionális mezőt gyakran nevezik attributumnak.

- Koncepcionális csoport: Zéró vagy több koncepcionális mező, és/vagy koncepcionális csoport kollektívája. A koncepcionális csoport tartalmának nem kell diszjunktnak lenni más koncepcionális mező vagy tartalmához viszonyítva. Valamely koncepcionális mező vagy csoport típus, illetve előfordulás zéró vagy több koncepcionális csoport-

hoz tartozhat.

- Koncepcionális rekord: (entitás rekord) : Zéró vagy több koncepcionális mező és/vagy csoport kollekciója, amely egy entitást reprezentál. Tartalmának nem kell diszjunktnak lenni más koncepcionális rekordok tartalmával szemben. Valamely koncepcionális mező vagy csoport típus, illetve előfordulás több koncepcionális rekordhoz is tartozhat.

A koncepcionális rekord tartalmazza azt a koncepcionális adatot, amit egy adott entitásról ismert tényeknek tartunk. Ezeknek az entitásoknak identitásuk van, ezért minden koncepcionális rekordnak azonosítója van (értelemszerű mennyiség, vagy önkényesen hozzárendelt rendszer azonosító).

Definiálható koncepcionális rekordok tetszőleges konstrukciója is. Egy koncepcionális rekordban szereplő összes, vagy néhány koncepcionális mezőtípus kapcsolatba hozható, ami ezután másik entitást (típust) reprezentál. Ha az így létrehozott konstrukciókat névvel látjuk el, akkor önálló koncepcionális rekord típusá válnak.

A koncepcionális mezőtípus, ami valamely koncepcionális rekordban azonosító volt, nem lehet azonosító, egy vele összefüggésben lévő másik rekord típusban. Ily módon ugyanaz a koncepcionális adat definícióval kialakított kapcsolatok hálózata lehet és több koncepcionális rekord formájában jelenhet meg. Bizonyos kapcsolatok, szándékosan, vezetői megfontolások miatt definiálás nélkül is maradhatnak.

A koncepcionális rekordok tetszőleges bonyolultságúak lehetnek (pl. IMS adatbázis rekord megfelelői). Tartalmazhatnak metszet koncepcionális adatot, egymással összefüggő vagy redundáns koncepcionális adatot, ami javíthatja az információ használhatóságát vagy értelmezését, ha az nem világos vagy egyértelmű. Defi-

niciója a címezhetőségre való tekintet nélkül történhet, azaz leíró tulajdonságait a teljes struktúrára adjuk meg szegmentálásra való utalás nélkül. Ismétlődő csoportokat nem különböztetünk meg ismétlődő szegmensektől vagy más elemektől, hanem ezeket összefoglalóan ismétlődő koncepcionális csoportnak nevezzük. Általában megengedett, hogy a koncepcionális objektumok leíró információja beágyazható legyen másokba, azaz bonyolultabb objektumok definiálhatók legyenek egyszerűbbek struktúrája és asszociációja révén. A legkevésbé bonyolult koncepcionális rekordokat ne jelenítsük meg a felhasználó felé és ezek ne legyenek közvetlenül összekötve externális rekordokkal vagy plexekkel. A vállalati adminisztrátor számára ajánlatos, ha ezeket a koncepcionális rekordokat harmadik normál alaku relációkként tartja fenn.

A koncepcionális rekord tipussal kapcsolatban fontos elv (amit kihasználunk, de be is tartunk), hogy definíciója stabil. Ha definiálásra került, addig él, amíg felhasználása meg nem szűnik és mivel közben nem kerül módosításra, már létező alkalmazóit nem zavarjuk (elvileg bővithető, kiegészíthető a létező felhasználók megzavarása nélkül). Ha akár csak kis mértékben is más definíció kívánatos valamely új alkalmazáshoz, vagy lényeges módosítás a már létezőkhöz, akkor újabb koncepcionális rekord típust definiálunk (amit ugyanazon internális adat alapján kreálhatunk), fenntartva változatlanul a korábbi koncepcionális rekord definíciókat. A környezet struktúrájában bekövetkező változást (vállalat szervezete; üzlet politikája; pénzügyi, ellenőrzési, számlázási elvek; működési szabályok, stb.) úgy modellezzük, hogy új koncepcionális rekord típusokat és hozzájuk tartozó leképzéseket definiálunk, amik szükségesek a változások követéséhez, vagy modellezéséhez. A létező koncepcionális rekordoknak internális rekordokból való képzését megvalósító leképzések akkor

kerülnek módosításra, ha a környezeti változás az internális modellben és a koncepcionális modellben egyaránt változást eredményezett. A létező, bizonyos feladatokhoz kijelölt vagy azokhoz rendelt koncepcionális rekord típusokat nem szabad módosítani, ha az internális adatból történő leképzés konstruálható oly módon, hogy a létező alkalmazások érvényessége fennmarad. Ugyanakkor szükséges lehet fenntartani a leképzési generációkat, hogy áttekinthessük történelmileg az internális adatot, hogyan szemléltük előbb és milyen változásokon ment keresztül.

Kíváncsi lehet, hogy a koncepcionális rekordokat kanonikusan definiálhassuk. Azaz képes legyen a rendszer ilyen deklarációk elfogadására és a koncepcionális séma processzornak meglegyen a képessége, hogy megjelenítsen leíró sajátságokat bármely számú ekvivalens strukturában. Másrészt az internális/externális transzformációt végző funkciónak legyen képessége megjeleníteni egy "előfordulást" (ténylegesen egy hozzá kapcsolt externális rekordot) valamely alkalmazói tárgy programhoz bármely számú ekvivalens strukturában. Nyilvánvalóan lehetővé kell tenni, hogy a koncepcionális processzor korlátozható legyen a koncepcionális rekordokra vonatkozó információ megjelenítésénél. A leírások csak felhatalmazott személyek felé adhatók meg, másrészt az internális/externális transzformációs funkció csak felhatalmazott tárgyprogramokhoz és előírt formátumban, strukturában adhat át információt. Ha kanonikus definiálásra nincsen eszköz létrehozva, akkor valamennyi koncepcionális rekordot explicite definiálni kell minden strukturához, amiben megjeleníthető és amihez hozzá rendelhető.

A koncepcionális rekordot gyakran entitás rekordnak is nevezik. Másik jelentése (ha hozzáértjük a nem strukturált, harmadik normál alakú formát) közbelső

adat objektumok definiálásánál és megvalósítási folyamatokban van.

- Koncepcionális plex: Zéró vagy több koncepcionális rekord és/vagy plex kollekciónja. A koncepcionális plex tartalmának nem kell diszjunktnak lenni más koncepcionális plexek tartalmához viszonyítva. Valamely koncepcionális rekord, vagy plex típus, illetve előfordulás jelen lehet zéró vagy több koncepcionális plexben is. Mivel a koncepcionális rekordok lehetnek olyan bonyolultságúak, mint több koncepcionális plex, továbbá, mivel a koncepcionális modellben a cimezhetőség nem követelmény, a definiáló izlésére van bízva, hogy valamely koncepcionális adatot koncepcionális rekord vagy plex formájában ad-e meg.
- Koncepcionális rekord-halmaz: Zéró vagy több koncepcionális rekord és/vagy plex kollekciónja, ami entitás halmazt reprezentál. A koncepcionális rekord-halmaz tartalmának nem kell diszjunktnak lenni más koncepcionális rekord-halmaz tartalmával szemben. Egy koncepcionális rekord, vagy plex típus vagy előfordulás több koncepcionális rekord-halmazban is szerepelhet. A koncepcionális rekord-halmaz definiálható másik koncepcionális rekord-halmazok rész-halmazain, egy vagy több koncepcionális rekord-típus összes előfordulásán, bizonyos előfordulásain, stb.

A koncepcionális rekord halmazok azt a koncepcionális adatot tartalmazzák, ami entitás halmazok bizonyos ismert tulajdonságait jelenti (nem szükségszerűen hasonló entitásokét). Lehet korlátozni a szokásos módon a halmazhoz tartozó sokaságot, megadható rendezésének elve, stb. Ez az a koncepcionális objektum, amihez egy externális rekord-halmaz kapcsolható. Gyakran nevezik ezt az objektumot, entitás rekord-halmaznak is.

- Konceptcionális adatbázis: Konceptcionális rekord-halmazok diszjunkt, integrált, névvel ellátott, egyértelmű kollekciója, amelyeket egy konceptcionális sémában irtunk le. Ez tartalmazza mindazt a konceptcionális adatot, amit a vállalatról ismerünk és tények formájában definiáltunk és itt reprezentáltunk. Szorosan összefügg egy internális adatbázissal annyiban, hogy a konceptcionális sémában definiált minden konceptcionális rekord-halmaz csak olyan internális rekord-halmazra hivatkozik, ami az internális sémában definiálva van, továbbá az internális sémában definiált minden internális rekord-halmazra csak olyan konceptcionális rekord-halmazból történik hivatkozás, amelyek egy konceptcionális sémában vannak definiálva. Így valamely konceptcionális rekord-halmaz teljesen beletartozik egy konceptcionális adatbázisba.

A konceptcionális sémában lehetnek olyan konceptcionális rekord-halmaz deskriptorok, amelyek nem hivatkoznak egyetlen internális rekord-halmazra sem a vonatkozó internális adatbázisban, ami jelzi, hogy az ehhez tartozó internális adat nincs még felállítva. Lehetséges az is, hogy egy konceptcionális rekord-halmaz úgy van definiálva, hogy duplikálja egy másik konceptcionális rekord-halmaz definícióját, annak jelölésére, hogy az ehhez tartozó internális rekord-halmaz másik internális adatbázisban létezik. Sem a konceptcionális séma processzor, sem az ABKR nem tudja diagnosztálni ezt a szituációt és nem tud nyújtani automatikus funkciót vagy irányítást ezen nyilvánvalóan nem létező internális adatra vonatkozóan.

2.5. Externális modell

Az externális modell objektumok olyan kollekciója, amely valamely alkalmazás, vagy alkalmazás család számára fontos entitásokat reprezentálja. Azt az objektumot, ami az entitások modellje, figyelembe véve az egyes alkalmazásokhoz (vagy alkalmazási családokhoz) való adaptálódást, itt externális rekordnak nevezzük. Az entitás ezen modelljét a COBOL-ban logikai rekordnak, adatstruktúra halmazban tulajdonos vagy tag rekordnak, relációkban sornak, kimeneti riportokban sornak, vagy bármi másnak nevezhetjük, ami illeszkedik a vonatkozó alkalmazás fogalomköréhez. A vállalaton belüli minden alkalmazás a másiktól különböző externális modell létrehozását kívánhatja. Célszerű biztosítani, hogy a különböző externális modellek származtathatók legyenek egyetlen közös, kanonikus externális modellből.

A különböző alkalmazások tervezhetőek externális modellek kizárólagos (nem osztott) vagy szabályozott módon közös használatára. Az externális modellekben szereplő objektumok az alkalmazás követelésére lesznek létrehozva és megszűnnek létezni, ha az illető alkalmazás számára tovább nem fontosak. Az általános adatbázis rendszer absztrakt externális modellekre hivatkozik, de az externális objektumok együttese valóságosan létezik, még ha megvalósulásuk nem is teljes minden időpontban.

Az externális modellben definiált objektum halmaz magába foglalja mindazokat az objektumokat, amelyek szükségesek egy kanonikus externális modell létrehozásához. Ehhez a modellhez tartoznak azon komponensek és azok halmazai (pl. attributum, sor, reláció, elemi tétel, csoport, logikai rekord, file, bejegyzés, riport részlet, tábla, riport), amelyek externális adatstruktúrákhoz, procedurális alkalmazói programozó nyelvekhez (COBOL, FORTRAN, stb.), speciális alkalmazásokhoz (ill. azok családjához) illeszkedően kerülnek definiálásra, vagy ezeket a konstrukciókat közelítik.

- Externális mező: Az externális adatobjektumok között ez a legkisebb elem, amit valamely alkalmazói program névvel referálhat. Mérete, dimenziója, mértékegysége és számos más nem-quantitativ interpretálása lehetséges. Lehet rögzített vagy változó hosszúságú.

Nem szükséges, hogy az externális mezők és az internális mezők értékei között 1:1 megfelelés legyen. Az externális mezők értéke eltolás, transzformáció, összekapcsolás, logikai vagy aritmetikai számolás vagy más algoritmus eredménye lehet, amit egy vagy több internális mező értékéből kaptunk, vagy lehet objektum előfordulások száma. Egy internális mező részt vehet egy vagy több externális mező létrehozásában is.

- Externális csoport: Egy vagy több externális mező és/vagy externális csoport kollektiója. Az externális csoport tartalmának nem kell diszjunktnak lenni más externális csoportok tartalmával szemben. Az externális mező (illetve csoport), vagy azok előfordulásai zéró vagy több externális csoportban szerepelhetnek.
- Externális rekord: Az externális rekord zéró vagy több externális mező és/vagy csoport kollektiója olyan formában, ahogyan azt az alkalmazói programozó látja. Konkurens hozzáférés van biztosítva externális rekordokhoz elemi adatmanipuláló műveletekkel. Az externális rekordok tartalmának nem kell diszjunktnak lenni más externális rekordok tartalmával szemben. Valamely externális mező vagy externális csoport típus, vagy előfordulás beletartozhat egy vagy több externális rekordba is.

Definíció szerint egy externális rekord minden externális mezője ugyanazon entitásra reprezentál tényeket. Minden entitásnak identitása van, így minden externális

lis rekordnak szükségszerűen azonosítója létezik. Ha az externális rekord változó számú, beágyazott ismétlődő externális csoportot tartalmaz, akkor hierarchikus struktúrájúnak nevezzük. Ha az externális rekord több externális plex részét képezi, akkor hálózati rekordnak nevezzük. Egy externális rekord hierarchikus és hálózati is lehet egyszerre.

Az ABKR rugalmasságától függően nem szükséges 1:1 megfelelésnek lenni internális rekordok (típus és előfordulás) és externális rekordok (típus és előfordulás) között. Valamely externális rekord-halmazbeli externális adat létrejöhet különböző internális rekordhalmaz részekből vertikális és horizontális összekapcsolással. Vagyis externális rekordok valamely sorozata (amely externális rekord-halmaz része) képezhető különböző internális rekord-halmazokhoz tartozó internális rekordok sorozatából (vertikális) és az egyes externális rekordok létrehozhatók különböző internális rekordok internális mezőiből (horizontális). Attól függően, hogy a konfliktív hozzáférés feletti irányítást hogyan oldja meg az ABKR, több mint egy externális rekord-halmaz lehet nyitva ugyanazon internális rekord-halmazon.

Valamely externális rekordtípus egy előfordulása létrejön a beletartozó externális mezők értékének létrejöttével abban az időpontban, amikor az elemi adatmanipuláló utasítás végrehajtása ezt megkívánja. Egy alkalmazás fenntartja érdeklődését és a konkurens hozzáférési jogát több mint egy externális rekordhoz ugyanazon vagy különböző externális rekord-halmazokban. Ha az externális alkalmazás számára nem érdekes tovább felszabadítjuk a rekordot, helyettesítjük másikkal, stb. (Ez ignorálja az ABKR bemenetnél előre, kimenetnél hátra való bufferekrolási funkcióját.)

Az externális rekordok feletti exkluzív irányítás ugyanazon vagy különböző externális rekordhalmazokban és a különböző programok közötti szinkronizálás következményei ugyanazok, mint a teljes adatbázis szintjén. A változások tovaterjedése (és tovább vitele) az externális rekord egy példányától másikhöz, ami ugyanabban vagy különböző externális rekord-halmazban van, hasonló e probléma adatbázis szintű megoldásához ugyanazon vagy különböző programok általi használatánál.

Egy irányított adatbázis rendszerben az externális rekordok és a bennük foglalt externális csoportok és mezők megengedhető kollekciói előre determináltak a koncepcionális sémában a koncepcionális rekordok leíró jellemzői formájában. Ahhoz, hogy egy externális rekord megengedhető legyen, valamelyik koncepcionális rekord rész-halmazának kell lennie (nem szükségszerűen valódi). Az irányítás alternatív technikája előre determináltság a koncepcionális sémában a koncepcionális rekordok komponenseinek megengedhető kombinációira. Mivel azonban koncepcionális rekordok konstrukciói szintén koncepcionális rekordokká válnak, ha névvel látjuk el őket, ez az alternatíva valójában az előbbi állítást jelenti más módon kifejezve.

Elvárás az adatbázis környezetében, hogy externális rekord típusok definiálhatók legyenek kanonikusan. Azaz képes legyen ilyen deklarációt elfogadni a rendszer és az externális séma processzornak legyen olyan képessége, hogy leíró tulajdonságokat bármely más nyelvhez vagy alkalmazáshoz orientált formában tudjon megjeleníteni. Továbbá az externális-koncepcionális jellemzőket összekapcsoló funkciónak legyen képessége egy externális rekordot bármelyik koncepcionális rekordhoz kötni annak ellenére, hogy az így létrejövő externális rekord megjelenítési formátuma eltérhet a koncepcionális rekord megjelenítési formátumától. Nyilvánvalóan szüksé-

ges lesz és az externális séma processzort lehet úgy kialakítani, hogy az externális rekordok leíró jellemzőit csak olyan személy számára jelenítse meg, akinek ehhez felhatalmazása van. Ugyiszintén az externális/konceptcionális jellemzők összekapcsolását végző funkció is csak felhatalmazott személytől vagy formátumban fogad el ilyen követelést. Amennyiben externális rekordtípusok kanonikus definiálására nincs lehetőség, akkor valamennyi externális rekord típust külön kell definiálni minden alkalmazáshoz és formátumhoz úgy, ahogyan a megjelenítés, illetve az externális/konceptcionális összekötést végző modul megkívánja.

- Externális plex: Az externális plex nulla vagy több externális rekord és/vagy externális plex kollekciója. Az externális plex tartalma nem kell hogy diszjunkt legyen más externális plexek tartalmához viszonyítva. Valamely externális rekord vagy plex típus vagy annak előfordulása zéró vagy több externális plexhez tartozhat. Bár az externális plexek előfordulásai valójában nem jönnek létre teljességükben (a beletartozó externális rekordtípusok előfordulásai létrejönnek egymás után, de referálásuk megszűntével törlődnek), definíciójuk olyan, mintha léteznének és a felhasználó létezőnek gondolhatja őket létrehozásukkor és manipulálásukkor.

- Externális rekord-halmaz: Zéró vagy több externális rekord és/vagy externális plex kollekciója olyan formában, ahogyan a felhasználó látja. Az externális rekord-halmaz tartalma nem kell, hogy diszjunkt legyen más externális rekord-halmazokhoz viszonyítva. Valamely externális rekord vagy plex típus, illetve ezek előfordulása beletartozhat egy vagy több externális rekord-halmazba is.

Egy externális rekord-halmaz definiálva lehet más externális halmazok rész-halmazain; egy vagy több externális rekord típus bizonyos előfordulásain, stb.

Az externális rekord-halmaz lehet nem-rendezett, amikor nincs explicit kapcsolat az externális rekordok között, ami sorrendet határozná meg. Az externális rekordok ekkor azonosítójuk szerint kerülnek kiválasztásra, vagy előre nem jósolható meg sorrendiségük. Másrészt az externális rekord-halmaz lehet rendezett, amikor van explicit kapcsolat az alkotó elemek között és ezt az alkalmazott rendezési elv reprezentálja, vagy az alkotó rekordok, illetve plexek strukturája, vagy valami más. Módosítva a strukturaképzésben résztvevő valamelyik externális mező értékét (ami irányítja a rendezést, vagy kialakítja az externális rekord-halmaz strukturáját) ez módosulást eredményez a vonatkozó externális rekord(ok) elhelyezkedésében is valamennyi érintett externális rekord-halmazban. A későbbiekben, legkésőbb az ABKR implementálásakor rögzíteni kell, hogy a változás hatása időben mikor lesz átvive olyan mezőkre és egyéb strukturákra, amit a strukturaképző vagy abban részt nem vevő mezők érintenek. Az átvitel természetesen azokra az externális mezőkre terjed ki, amelyek közösen használják a vonatkozó (és a változással érintett) internális mezőt, de az átvitel időpontja sok tényezőtől függhet.

Az externális rekord-halmazok használatának kellékei (pl. felhasználási címkék) láthatók az alkalmazások oldaláról is, de az internális adattárolási komponensek (pl. kötet címkék, indexek, pointerok, hash-cimek) rejtve maradnak az internális modellben a felhasználó előtt.

Bár az internális rekord-halmazok nem realizálódnak teljességükben egyidejűen (a rekordok realizálódnak egyenként, de meg nem szűnnek) definíciójuk olyan, mintha léteznének és felhasználói szemléletük ilyen lehet. Az externális rekord-halmaz abban az időintervallumban létezik, ami megnyitásától lezárásáig tart. A realizálásukban résztvevő internális adat megnyitásuk előtt és lezá-

rásuk után is létezik internális rekord-halmazbeli internális mezők formájában.

2.6. Internális modell

Az internális modell olyan objektumok kollekciója, amelyek a tárolt adatot tartalmazzák és reprezentálják az externális és koncepcionális modellek objektumait. Az internális modell motivációja eltér az előző két modell motivációjától: a számítógépes eszközök és facilitások gazdaságos használatát igyekszik biztosítani úgy, hogy a feldolgozási követelményekkel konzisztensek maradjunk.

Az internális modell kialakításának (architektúrájának) szabad és kell tükröznie a jelenlegi tárolási technológiákat, valamint a leképzendő koncepcionális modellt. Alternatív internális modellek lehetségesek, amelyek más gazdasági környezethez, konfigurációhoz, implementálási szempontokhoz vagy összeállítási meggondolásokhoz illeszkednek. Lehetőségeket kell biztosítani arra, hogy további objektumtipusok legyenek illeszthetők a választott internális modellhez, a már létezőkhöz hozzá tudjunk adni, törölni belőlük, stb. Ezek a kiterjesztések, bővitések további interface-k és facilitások rendelkezésre állását kívánhatják meg és nem szükségképpen vesznek részt az internális modell és a koncepcionális modell objektumai közötti leképzésben. Ilyen természetű kiterjesztések példái a különböző hozzáférési módszerek, adat-típus függő adatbázis facilitások, olyan irányítási és flexibilitást adó eszközök, amiket a koncepcionális séma által működtetett ABKR adhat. Bár ezeket az alternatív lehetőségeket nem tekintjük architektúrálisan szükségesnek az adatbázis rendszerben, ezek léteznek a mai ABKR implementációkban, és alkalmazásukra gondolni kell.

Az internális modell objektumai léteznek, permanensek abban az értelmezésben, hogy ha egyszer fizikailag

tárolásra kerültek az adatbázisban, akkor ott maradnak explicit, fizikai eltávolításukig. Az internális objektum sokaság létezik ezen időintervallumban.

- Internális modell tér: Annak a cimtérnek az absztrakciója, amiben az internális rekordokat tároljuk. Az internális modell teret az internális séma céljaira egy sík, nem korlátos, többorigós, lineáris cimtér reprezentálja, amelyben az elmozdítás egysége bit, byte, szó, internális rekord, fizikai rekord (internális vagy externális tároló rekordok), nyom, sáv, cylinder vagy kötet, stb. lehet. A rendszer kontrol adata, melyek egy kötetben elhelyezett adatok (tartalomjegyzék, directoryk, kötet címkék stb.) láthatók az internális modellben. Az internális adattároló szervezésének működés orientált jellemzői (pl. valamihez közeli elhelyezés, más egységen keresztüli hozzáférés, indexek vagy vezérlő blokkok többszörös példányai, internális adat redundáns példányai (back-up, célhoz igazított vagy szétosztott)) érzékelhetők az internális modellben. A külső tároló közeg működés orientált jellemzőit (pl. kötet kapacitás, nyom hosszúság, látensség) az internális modell belső adattároló szervezésében tükröztetjük, viszont annak fizikai jellemzői (bit reprezentálás, redundancia vagy paritás ellenőrzés, adatátvihetőség vagy csere szempontjából érdekes megfontolások) nem láthatók az internális modellben.
- Hely terjedelem: Monoton növekvő címszámok összefüggő szub-allokációja a cimtérben, ami zéró vagy több internális rekordtípus előfordulásait tartalmazhatja. Beletartozhat egy vagy több forma terjedelemben (vagy azok részeibe) előforduló internális rekord. A hely terjedelem definíciója nem foglalja magába annak leképázését bármiféle külső tároló közegre.

Egy hely terjedelem tartalma diszjunkt más terjedelmek tartalmával szemben. Az internális rekordok vagy mezők előfordulásai mindig valamely hely terjedelemben vannak,

de az internális mező vagy rekord típusok előfordulhatnak több hely terjedelemben is. Egy internális mező vagy rekord előfordulása nem szükséges hogy teljes legyen abban a hely terjedelemben, amihez tartozik.

- Forma terjedelem: Internális rekord-halmaz olyan al-szakasza, ami egy vagy több internális rekord típus zéró vagy több előfordulását tartalmazhatja. Minden itt előforduló internális rekord típusnak ugyanazon internális rekord leíró jellemzői kell hogy legyenek. Egy forma terjedelemben az internális adatobjektumok tulajdonságai ugyanazok az egység minden előfordulásához. Ha valamely tulajdonság megváltozik (pl. rögzített lebegőpont helyett skálázott megjelenítés egy internális mezőtípust eltávolítunk az internális rekord típusból; a reprezentált mező virtuális az eddigi redundáns helyett), akkor új leírás halmazt kell kreálni ugyanazon internális rekord típushoz. Ha bármely internális rekord leíró jellemző megváltozik, új forma terjedelem jön létre. Így ugyanazon internális rekord-halmazban ugyanazon rekord típus konkurrensen különböző leírásokkal rendelkezhet a különböző forma terjedelmekhez. Ugyanazon hely terjedelemben egy vagy több forma terjedelem lehet, vagy ugyanazon forma terjedelem egy vagy több hely terjedelemben is beletartozhat.

Egy forma terjedelem tartalma diszjunkt más forma terjedelmek tartalmával szemben. Egy internális mező vagy rekord egy előfordulása valamely forma terjedelemben van, de egy internális mező vagy rekord típus lehet egy vagy több forma terjedelemben. Internális mező vagy rekord egy előfordulása nem kell, hogy teljes legyen abban a forma terjedelemben, amihez tartozik.

Az előbbi három internális modell objektumnak leképzés szempontjából ekvivalens megfelelői az adattárolási modellben találhatók meg. Az externális vagy koncepcionális séma objektumai 1:1 kapcsolatba nem hozhatók ezekkel

objektumokkal viszont a többi internális modellbeli objektum előfordulásai "konténereinek" tekinthetők (rajtuk keresztül az externális és koncepcionális séma objektumai hordozóinak is tekinthetők). A következőkben ismertetendő internális adatobjektumok nagyobbik része közvetlenül is kapcsolható a megfelelő externális vagy koncepcionális sémabeli objektumhoz (bár nem mindig 1:1 kapcsolat van), kisebbik része viszont az internális adatok szervezéséhez célszerűen kialakított adat objektum.

- Internális mező: (adatértelem): ez az internális modell legkisebb névvel ellátott internális adat objektuma. Értéke az internális modell legkisebb egységeinek egymás utáni sorozatába (stringjébe) kerül tárolásra bekódolt formában. Nagysága, dimenziója, mértékegysége és egyéb nem-kvantitatív interpretációja is van, továbbá lehet rögzített vagy változó hosszúságú (modelltér egységeiben mérve). Az externális mezők értéke transzláció, transzformáció, konkatenálás, logikai vagy aritmetikai számolás, egy vagy több internális mező értékén végrehajtott algoritmus eredménye lehet vagy bizonyos objektumok előfordulásainak száma. Egy internális mező részt vehet egy vagy több externális mező értékének létrehozásában is.

(Az internális mező fogalmat gyakran adatelemnek nevezik. Ebben a riportban adatelemnek olyan elemi tárolt adatobjektumot nevezünk, amelyet nem integráltunk az adatbázisba.)

- Internális mező aggregátum: internális mezők lehetnek aggregáltak miközben alárendelt részei lesznek internális rekordnak és tükrözik az internális adathoz való hozzáférés mechanizmusát. Az internális mező aggregátum gyakran kongruens a koncepcionális rekord egy részével, ha ilyen részeket bizonyos alkalmazások sorozatosan vagy konkurrensen igényelnek. Az internális sémában nincs törekvés arra, hogy ilyen adataggregátumokat külön definiáljunk, jellegmezzünk vagy névvel lássunk el.

- Internális rekord (tárolt rekord): egymással kapcsolatban lévő, egyértelműen azonosítható sorbakapcsolása internális modellbeli internális mezőknek. Az internális rekord előfordulás összes internális mező értéke az internális modell tér egységeinek összefüggő láncolatában kerül tárolásra. Az internális rekord előfordulások tartalma diszjunkt más (azonos típusu) internális rekord előfordulások tartalmával szemben. Ugyanakkor egy internális rekordtípus tartalmának nem kell diszjunktak lenni más internális rekordtípusok tartalmával szemben, azaz egy internális mezőtípus egy vagy több internális rekord típushoz tartozhat.

Az internális rekordhoz tartozó internális mezők közötti kapcsolatok tetszőleges fajtajuk lehetnek. Az internális mezők valamely entitásra vonatkozó különböző tényeket reprezentálhatnak, entitások kiválasztására vonatkozó tények csoportját vagy más kombinációit. Ezeket a kapcsolatokat az internális adattároló szervezés szemszögéből nézve specifikáljuk, azaz inkább tükröznek praktikussági és gazdaságossági megfontolásokat mint információsakat (inkább fizikai, mintsem logikai tartalmuk).

Az internális rekordot gyakran tárolt rekordnak hívjuk (ami azonban általában nem ekvivalens a "fizikai" rekorddal) mivel az általános szóhasználat szerint ez magába foglalja a tényleges tárolást rekord aggregátum (lap, blokk, stb.) tagjaként tároló közegen, annak valamely kötetében. (Itt mi tárolt rekord alatt adatelemek olyan kollekciónak értjük, ami nincsen integrálva internális adatbázisba.)

- Internális rekord aggregátum: internális rekordokat blokkokba, lapokba, stb. aggregáljuk, hogy tükrözzük a rendszeren belüli hozzáférési, elérési stratégiát. Ezt a stratégiát befolyásolja az internális modell tér kezelése, indexelése

si mód, a tároló látensség, a hivatkozások sorrendiségi és konkurencsségi sajátosságai. Internális rekord aggregátumok gyakran konvergensenkoincidálnak koncepcionális rekordokkal vagy plexekkel, illetve azok részeivel, mivel ezeket egyes alkalmazások teljességükben referálják. Ugyanakkor nincs törekvés az internális sémában, hogy ezeket az internális rekord aggregátumokat külön megnevezzük, definiáljuk és más önnálló jellemzőkkel lássuk el.

- Internális rekord halmaz (adat halmaz): az internális rekord halmaz kollekcióna zéró vagy több előfordulásának egy vagy több internális rekord típusnak, ami összefügg egy egyedi címzési, elérési séma rendszerrel és ezért felmutat közös internális adattároló szervezést. Az internális rekord-halmazok tartalma diszjunkt más internális rekordhalmazok tartalmához viszonyítva. Egy internális rekord előfordulás teljes egészében része a vonatkozó internális rekord halmaznak. Ugyanakkor egy internális rekord típus egy vagy több rekord halmazhoz is tartozhat. Egy internális rekord halmaz egy vagy több hely terjedeleme és forma terjedeleme kollekcióna és egyben a legnagyobb adat objektum, amihez másik objektum kapcsolható.

Az ABKR rugalmasságától függően nem szükséges 1:1 megfelelés internális és externális rekord-halmazok között. Az externális rekord-halmaz tagjai különböző internális rekord-halmazokból képezhetők vízszintes és függőleges konkatenálással. Vagyis externális rekord-halmaz externális rekordjainak sorozata létrehozható különböző internális rekord-halmazokhoz tartozó internális rekordokból (vertikális konkatenálás) és az egyedi externális rekordok tartalma származhat különböző internális rekordok mézöiből (horizontális konkatenálás). Attól függően, hogy hogyan irányítja az ABKR a konfliktív hozzáférést, egyidejűen több mint egy externális rekord-halmaz lehet nyitva ugyanazon internális rekord-halmazokon. (Az internális modell belső adattároló szervezésének elemei (pl.kötet

cimkék, internális rekord-halmazhoz tartozó címkek, indexek, pointer tömbök, hash címző algoritmus specifikációja.)

Internális rekord-halmazokat gyakran "adat-halmaz"-nak is nevezik. Ez nem ugyanaz, mint a fizikai file, mivel az a hordozó közeget, annak kötetét és nem a tárolt adatot foglalja magába. Itt az adat-halmazt olyan tárolt rekord kollekcióként értelmezzük, ami nincs integrálva az internális adatbázisban.

- Internális adatbázis: egyszeri, diszjunkt névvel ellátott kollekciója internális rekord-halmazoknak, amit egy internális sémában irtunk le. Általában a vállalat célirányosan kezelt, számítógéppel feldolgozható része a vállalati adatbanknak és a ténylegesen tárolt adatot jelenti.
- Adatbank: a vállalatnál ismert adatok teljes kollekciója. Részünkre csak az operatív, számítógéppel olvasható adatot jelenti, ami on-line és off-line internális rekord-halmazok együttese és olyan adat-halmazoké, amik felszerelhető és fel nem szerelhető kötetekbe vannak feljegyezve. Tartalmaz olyan internális rekord-halmazokat, amik egy vagy több internális adatbázisba vannak integrálva, továbbá olyan adat-halmazokat alkothatnak, amik nincsenek integrálva egyik internális adatbázishoz sem.

Lehetnek olyan adat-halmazok, amelyek nem függnek össze egyetlen internális adatbázissal sem, mivel nincsenek definiálva internális sémában, de az ABKR kezelése és irányítása alá tartoznak. Ilyen adat-halmazok példái a forrás adatok sorai, kimenő adatok sorai (lapjai), a virtuális memória lapjai, scratch adatok, check pointok adatai, stb.

2.7. Adatmodellek és birodalmak kapcsolata

Az előbbiekben leírtuk az egyes birodalmak és modellek lényeges sajátosságait, objektumaikat, fogalmaikat. Több helyen utaltunk arra, hogy a valóság világ entitásainak milyen szimbolikus megjelenései lehetnek az egyes modellekben. Érdekes azonban összefoglalóan is megadni az egyes birodalmak közötti kapcsolatokat, megfeleléseket.

Lényeges sajátossága az adatbázis vezérlő rendszernek, hogy externális modellek kivonhatók legyenek másik externális modellből, másrészt a koncepcionális modellből, továbbá, hogy leképezés konstruálható a koncepcionális és az internális modell között. Nem szükséges, hogy a három modell fajta ugyanazt a valóságot reprezentálja teljesen. Valamely internális rekord-halmaz nem szükségszerűen reprezentál egy specifikus entitás-halmazt; az internális rekord mezői nem szükségszerűen ugyanazon entitásra vonatkozó tényeket reprezentálnak (például egy invertált file-ban az internális rekordok minden internális mezeje ugyanazt a tényt tartalmazza különböző entitásokhoz). Másrészt az internális rekord nem szükségszerűen 1:1 kapcsolatban van valamelyik definiált koncepcionális rekorddal.

Az ABKR rugalmasságától függően nem szükségszerű az 1:1 megfelelés internális és externális rekord-halmazok között. Valamely externális rekord-halmaz externális adata létrehozható ("materializálható") különböző internális rekord halmazokból vertikális és horizontális konkatenálással. Azaz az externális rekordok sorrendje egy externális rekord-halmazban létrejöhet különböző internális rekord-halmazok internális rekordjai sorozatából (vertikális konkatenálás). Az ABKR-ben a konfliktív hozzáférés feletti irányítás megoldásától függően ugyanazon internális rekord halmazon több mint egy externális rekord-halmaz lehet nyitva ugyanazon időpontban.

Táblázat

Valós világ	Externális modell	Koncepcionális modell	Internális modell
Egyezmények, szabályok, szokások	Externális séma	Koncepcionális séma	Internális séma
Vezetés	Alkalmazói adminisztrátor	Vállalati adminisztrátor	Adatbázis adminisztrátor
Vállalat		Koncepcionális adatbázis	Internális adatbázis
Entitás halmaz	Externális rekord halmaz	Koncepcionális rekord halmaz	Internális rekord halmaz
	Entitás plex	Koncepcionális plex	Lásd Megj.:
Entitás	Externális rekord	Koncepcionális rekord	Internális rekord
	Externális csoport	Koncepcionális csoport	Lásd Megj.:
Tulajdonság	Externális mező	Koncepcionális mező	Internális mező.

Megj.: A rekordon belüli és a rekordok közötti rekord-felépítés függ a belső adattároló szervezés hely és működés orientált döntésétől és nem az információs kapcsolatoktól. Így ezekre a strukturákra nem adható egyetlen általános, kanonikus kiterjesztés az internális modellben.

Jogos azzal a feltételezéssel élni, hogy az entitások és a koncepcionális adat (ami tényeket reprezentál entitásokról) hosszabb életű, mint azok a technikák és technológiák, amin a hardware és a software implementációk nyugszanak. Így az internális modell implementációjának képesnek kell lenni a változtatásra, ha a technológia és technika változik, hogy optimális, gazdaságos maradjon. Az alkalmazási követelmények bizonyos kombinációjánál gyakran gazdaságos, ha a modellek azonosak sőt a technológia és technika bizonyos állásánál előfeltétel is lehet a modellek azonossága.

Fenntartva azt az elvet, hogy a modellek különbözők lehessenek és képesek legyünk ezt az elvet végig tartani, átvinni más elvekhez is, szükséges olyan terminológiát definiálni és használni, amiből a modellek és birodalmak közötti különbségek kitűnnek.

A mellékelt Táblázat mutatja az eddigiekben definiált objektumok közötti megfelelést. Ahogy a legmagasabb szintű vezetéstől várják el, hogy a vállalat működését meghatározó stratégiai irányelveket megadja, úgy az adatbázis adminisztrátora határozza meg az internális sémát, amely szerint az internális adatbázis működik. Az entitás halmaz entitások kollekciója, az internális rekord-halmaz internális rekordok kollekciója, ami eltérő valami. A rekordon belüli és rekordok közötti konstrukciók erősen függenek hely és működés orientált adattároló szervezési megfontolásoktól (kevésbé az általuk reprezentált információs kapcsolatoktól) ezért egyetlen általános, kanonikus kifejezés az internális konstrukciókra nem alkalmazható, elmentében az externális és koncepcionális konstrukciókkal, ahol ilyen létezik.

2.8 Objektumok kötése és leképzése egymás között.

Az információs rendszer folyamataiban résztvevő entitások és tulajdonságaik reprezentálhatók:

- olyan objektumokkal, amelyek leíró jellemzőit a forrás programban deklaráltuk (pl. COBOL DATA DIVISION-ban)
- olyan objektumokkal, amelyek leíró jellemzőit relációs externális sémákban definiáltuk és amit a COBOL externális séma definiál
- olyan objektumokkal, amelyek kanonikus externális sémákban kerülnek definiálásra.

Az objektumok leíró jellemzői, amelyeket akár közvetlenül forrásprogramban, akár externális sémákban definiáltunk, össze vannak kötve olyan objektumokkal, amelyeket a koncepcionális sémában definiáltunk. Az ezen sémákban deklarált objektumok és leíró jellemzőik valamennyien absztrakta, viszont az internális adat ténylegesen tárolásra kerül azokban az objektumokban, amelyek leíró jellemzőit az internális sémában deklaráltuk. A koncepcionális sémában definiált objektumok leíró jellemzői leképeződnek azokhoz az objektumokhoz, amelyek leíró jellemzőit az internális sémában adjuk meg. Ez nem jelent azonban 1:1 leképezést a szóbanforgó objektumok között.

Olyan objektumok, amelyek lokálisak egy alkalmazás externális modelljéhez, lehetséges, hogy nincsenek reprezentálva az internális sémában, és ezek nincsenek a vállalati adminisztrátor vagy az alkalmazási adminisztrátor irányítása alatt. Ezesetben a koncepcionális sémában sincsen olyan definiált objektum, amihez az externális sémában használt lokális objektum hozzá lenne kötve.

A koncepcionális sémában lehet olyan objektum, ami nincs reprezentálva internális adattal. Ez szokott történni az adatmodellek kiterjesztése, továbbfejlesztése idején, vagyis akkor, amikor az internális adat definiálása és gyűjtése még nem történt meg, de a koncepcionális és az externális sémában definiálásra kerültek az új objektumok. Az ilyen redundáns leírás elősegítheti az adatbázisban tárolt adatok közötti kapcsolatok megértését, másrészt utalhat másik adatbázisban tárolt adatra vagy olyan adatokra, amelyek nincsenek az adatbázisba integrálva, vagy nem számítógéppel feldolgozható közegen/formában vannak tárolva. Ezesetben nincs tervbe véve, hogy az externális sémában definiált ilyen objektum hozzá legyen kötve a koncepcionális sémában definiált valamelyik objektumhoz (kivéve tesztelés, ellenőrzés esetét).

2.9. Objektum leírások kaszkádolása és megvalósítása

A megvalósítás ("materializálás") intuitíve azt jelenti, hogy valamely definiált externális rekordot tárolt (internális) objektumokból kivonunk és létrehozunk (ez a GET utasítás). Ugyanezzel a szóval jelöljük az ellenkező irányú műveletet is, amikor az externális rekord tartalmát tárolt objektummá alakítjuk (ez a PUT utasítás). A megvalósítás szóval mindkét műveletet jelöljük és a szövegösszefüggés határozza meg mit is jelent az valójában. Az esetek többségében internális modellből externális modellre történik a megvalósítás, de egy konkrét ABKR-nek foglalkozni kell externális modellről (externális rekordok bizonyos közegen) externális modellre (externális rekordok a felhasználó munkaterületén) történő megvalósítással is.

A konkrét implementációk a deszkriptorok átnézésében és materializálás tekintetében széles skálát foghatnak át és modellről-modellre különbözők lehetnek. A funkcionális flexibilitás valamint a működési jellemzők különböző trade-off-ot tesznek lehetővé az implementációk között. Az implementálás egyik véglete, ha megkivánjuk, hogy egy koncep-

cionális rekord kongruens legyen egy externális rekorddal. Ezesetben az externális rekordok közvetlenül, 1:1 megfeleléssel létrehozhatók a tárolt internális rekordokból. Nagyobb rugalmasságot biztosít egy implementáció, ha lehetővé teszi, hogy egy teljes koncepcionális rekord internális rekordokból legyen képezhető, továbbá megengedi, hogy az externális rekord ebből a koncepcionális rekordból legyen kivonatolható.

További implementációs lehetőség, hogy a koncepcionális rekordokat nem hozzuk létre mint közbenső eredményt, hanem csak a leíró jellemzőket manipuláljuk és az általuk leírt koncepcionális objektumokat nem. Kifejleszthetők olyan processzorok (transzformátorok), amelyek végigpásztázzák a leíró jellemzőket statikusan és leképzést generálnak (végrehajtható kódot kompilálnak), ami az externális rekordokat létrehozza közvetlenül az internális rekordokból. Másik lehetőség, hogy a leíró jellemzőket dinamikusan pásztázzuk végig lokálisan optimalizált materializáló algoritmusokkal. Így lehetőség van az internális rekordok forma terjedelmeként változó tulajdonságainak figyelembevételére is a materializálási folyamatban akkor, amikor a feldolgozás találkozik ezen objektumokkal.

Az implementációk spektrumának legrugalmasabb formája az, ha lehetővé tesszük a leíró jellemzők statikus és dinamikus végigpásztázását egyaránt és a rendszer határozza meg melyik módszer az optimális az illető alkalmazásra, vagy hozzáférési módra. Figyelembeveszi az objektumok illeszkedésének mértékét és azt előnyösen kihasználja, az externális rekordoknak az internális rekordokba való beleágyazottságát hasznosítja, továbbá a leíró jellemzők időbeli stabilitását és az illető művelethez átnézendő internális objektumok számát is bevonja a statikus/dinamikus materializálás kiválasztásának eldöntéséhez.

Az eddig leírt elvek a felsorolt implementációs spektrum valamennyi eleméhez alkalmazhatók és valószínű, hogy egy adatbázis rendszer teljes életciklusa során a materializálási folyamatok jellemzői többször változnak objektum csoportokra, vagy egyedi objektumokra.

2.9. Adatfüggetlenség

2.9.1. Az adatfüggetlenség jellemzése

Az információs rendszerek lényegi sajátása, hogy folyamatokat, eljárásokat kössön össze az adatokkal, amelyeknek feldolgozására hivatottak vagy amelyek szükségesek az eljárások révén realizált funkciók teljesítéséhez. Mivel mind az eljárások, mind az adatok a felhasználó részére érdekes szemantikai sajátosságait bizonyos szintaxissal fejezik ki, az egyik területen végbemenő szintaxis változáshatással van a másokra is. Ettől függetlenül szemantikai változások is fellelhetők, de ezek iránti tűrőképessége sokkal jobb az információs rendszer minden résztvevőjének, mint a szintaxis változásokból tovább gyűrűző változások esetén.

Az ABKR által nyújtott adatfüggetlenség így az a képesség, hogy elszigetelje a felhasználót a tárolt adattal való olyan kölcsönhatástól, hogy a szintaxis változás nagyon költséges legyen, továbbá, hogy a szemantikai változások átvezetése is gazdaságos, rendezett módon valósulhasson meg. Azok az előnyök, amik adatfüggetlenségi sajátságokból származnak jól érzékelhetők, sőt egyszerűsített szituációkban számszerűsíthetők is. Ami időnként zavart okoz a fogalom értelmezésénél és az ABKR ezen tulajdonságainak értékelésénél az az, hogy az adatnak több olyan sajátága van, amitől egy alkalmazás függetleníthető ugyan,

de ezen jellemzők által nyújtott adatfüggetlenség mértéke eltérő lehet és ezen funkciók időzítése sem közömbös. Ugyanakkor a dolgok természetéből és a megoldás gazdasági kihatásaiból következően az eljárások és adatok egymásrahatását nem lehet elkerülni. Az ABKR-nek lehetnek bizonyos képességei, hogy a különböző változások hatását automatikusan megoldjuk, de teljes adatfüggetlenség nem létezik, mivel mindig történhet olyan perturbáció az információs rendszer környezetében ami az alkalmazás eredeti szándékát erősen módosítja.

Szemponatok, amelyek az adatfüggetlenségi eljárásokat motiválják:

- objektumok reprezentálásának változtatása.
- tárolt adat elhelyezésének és/vagy formátumának változtatása
- az illető adatot hasznosító egyéb felhasználók (akik nem ismertek valamely alkalmazás szempontjából) követelése, amelyek nem kongruensek egymással
- ismeretlen felhasználókkal együtt történő konkurrens adat hasznosítás
- valamely program által használt adat véletlen vagy szándékoltszerű megértése más felhasználó által.

A tárolt adat integritása is az adat függetlenség függvénye, amit gyakran nem veszünk kellően figyelembe. A tárolt adatnak azonban sem használata, sem fenntartása, kezelése nem lehet független a vállalatnak az információs rendszerrel szembeni követelményeitől. A feldolgozó programok azonban (sokszor implicit elvárás vagy követelmény) nem lehetnek kitéve olyan hatásoknak, amik részükre externálisnak tekinthetők. Az adatfüggetlenség tehát az előbbiekben felsorolt környezeti változások és/vagy fejlődések hátrányos hatásaitól igyekszik elszigetelni a felhasználót.

Az adatfüggetlenség nem foglal magába olyan képességet az ABKR részéről, hogy automatikusan megoldódik a programbeli algoritmus változás vagy a programon belüli adat szemléletben bekövetkezett változás hatása, vagy megoldásra kerül a tárolt adat véletlen (vagy szándékolt) hiányából eredő helyzet. A programok nem mentesek belső logikájukban, funkciójukban, adatszempléletükben belsőleg bekövetkező változások hatásaitól. Az adatfüggetlenségtől máguasságot nem lehet várni. Az ABKR ezen tulajdonsága a tárolt adatnak és leírásának programbeli szabad referálását engedi meg, hogy elszigeteljük a programokat az adat környezetben végbemenő változásoktól és adat forgalmazási mechanizmustól, vagyis elsősorban az alkalmazott adattárolási módtól; az adat programok közti megosztásának megvalósításától; az adatbázis olyan jellegű átszervezésétől, ami egyes programok, vagy alkalmazási családok működési jellemzőit vagy a teljes rendszer gazdaságossági mutatóit javítják. Így sikerrel birkózhatunk meg az adattárolás szervezésének változásával a tárolt adatértékek elhelyezésében, reprezentálásában történő módosításokkal és maguknak az értékeknek a változásával. Ha a programok nem kapnak hátrányos effektusokat az előbbi, számukra externális eseményektől, akkor az ABKR adatfüggetlenségi tulajdonságai kielégítőnek mondhatók.

Az adatfüggetlenség általában nem explicit "szolgáltatás"-ként jelenik meg az ABKR-ben, hanem a tárolt adat alternatív szemlélete révén valósul meg.

Az ABKR lehetővé teszi és működése során megőrzi, rendelkezésre bocsátja ezeket a szemléleteket, ugyanakkor ad olyan lehetőséget, hogy közben másik szemléletet alakítsunk ki a már tárolt adathoz annak érde-

kében, hogy másik fajta (valószínűleg konkurens és az eddigivel konfliktust fenntartó) használatát biztosítsuk, vagy módosítsuk a tárolt adat jellemzőit, működési és egyéb sajátosságait, illetve tartalmát.

Az adatfüggetlenség szükségességét nem szabad elhanyagolni, és arra törekedni, hogy olyan kompatibilitást hozzunk létre és tartsunk fenn, ami biztosítani igyekszik, hogy minden változtatás és eltérő használat "kompatibilis legyen egymással". Nem bizonyos szabályok és kellő fegyelmezettség, hanem a kiahaszínlható rugalmasság jelenti az adatfüggetlenségét. Ez nem biztosítható a tárolt adat olymódon való "szervezésével", hogy sose kelljen változtatni. A változás és változtatás elkerülhetetlen, és az adatfüggetlenség nem a változás elkerülésének képessége, hanem a változás okozta következmények redukálásának képessége. A változások ilymódon való átvezetésében két fontos szempont veendő figyelembe:

- azon leképázések bonyolultsága, amikhez alkalmazkodni lehet
- a változások dinamikája

A leképázések bonyolultsága olyan dolgokat foglal magába, mint egy érték reprezentáció jellemzőinek változása (hossz, kinézet, stb.), változtatás internális mezők redundancia sajátágában vagy mező értéket előállító algoritmusokban, egyes rekord típusok előfordulásaihoz való hozzáférési mód változtatása (hash kódolás, indexek képzése, láncolás, stb.). A változás dinamikája olyan dolgokat foglal magába, mint a tárolt adat átszervezésének megtiltása ha egyszer programok által való használatát megengedtük; a tárolt adat egésze átszervezésének befejeződése az ismételt használat-hoz való rendelkezésre bocsátás előtt; részleges átszervezés megengedése, miközben a tárolt adat használatban marad; externális modellbeli módosítások és

kiegészítések megengedése változatlan internális modellel mellett; az externális modellek továbbfejlesztésének megengedése konkurrensen az internális modell fejlesztésével.

A koncepcionális séma elhelyezését az externális séma (vagy az externális séma valamilyen megtestesülése, ami összefügg egy végrehajtandó tárgy programmal, pl. egy "adatbázis vezérlő blokk", ami az adatbázis kezelő rendszernek egy funkció ellátása érdekében történő behívásánál aktivizálódik) és az internális séma közötti interpretációs utban az motiválja, hogy olyan indirekciós szintet kapjunk, ami (statikus vagy dinamikus) adatfüggetlenséghez feltétlenül szükséges. Ha elhagynánk a koncepcionális sémát az objektum realizálási utból, és közvetlenül hozzákötnénk egy externális sémabeli objektumot (nevét és jellemzőit) az internális sémában leírt sémabeli objektumhoz (ottani névhez és tulajdonságokhoz), akkor ennek hatása az adatfüggetlenségi tulajdonságokra előre becsülhető lenne. Nagyon nehéz elképzelni realizztikusan parametrikus, generikus alkalmazási programok írását, vagy olyan programokét, amik tudják élni az internális adat szükségszerűen bekövetkező változásait, az ABKR olyan szolgáltatásai nélkül, amik az internális adat jellemzőinek változását kellően megoldják.

A különböző funkciók és tulajdonságok hozzárendelése az információs rendszer software strukturájának különböző rétegeihez ("hagyma" számítógép) alkalmanként izlés dolga lehet, de a rendszer minőségi jellemzői tekintetében és vállalaton belüli feladatok helyes ellátásánál kritikus is lehet. Ha minden alkalmazást úgy tervezünk (vagy lehet úgy tervezni), hogy a tárolt adatoknak csak jól körülhatárolt, előre meghatározható tömegén dolgozik és ennek minden jellemzője

uniform és előre ismert, akkor az eljárás/adat statikus összekötése (pl. fordításkor) elegendő és a funkciók hozzárendelése valamint időzítése nem súlyos rendszertехnikai kérdés. Másrészt, ha valamely alkalmazást úgy (kell) terveznünk, hogy generikusan dolgozzon (alternatív, helyettesíthető adattömegeken vagy ezen adattömegek bármely jellemzője változhat) az ismételt feldolgozások vagy újra fordítások között, illetve a tulajdonságok nem uniformok a tárolt adattömegeken (forma terjedelmek léteznek) akkor a dinamikus eljárás/adat kötés képessége szükséges az ABKR-ben. Ez a képesség technikailag nagyon érzékeny az egyes funkcióknak a különböző processzorok és sémák közötti szétosztására és azok időzítésére. Például, ha az ABKR nem nyújtja azokat a fajta funkciókat (amik az operációs rendszer utasításaival általában biztosíthatók), hogy program végrehajtáskor nevezzük meg a használandó tárolt adattömegeket, akkor értelmetlen és gazdaságtalan parametrikus, generikus alkalmazásokat írni.

Kétséggel bőséges az olyan alkalmazások köre, amelyek statikus kötéssel kielégítően megoldhatók. Az ABKR-en belül, amely nyújt dinamikus kötési lehetőséget, a statikus kötésnek sem gazdaságossági, sem taktikai szempontból sem szabad háttérbe szorulnia. Kellő adatfüggetlenséget adó ABKR környezetben valamely alkalmazás folytathatja működését (esetenként redukált működési jellemzőkkel), ha a tárolt adatot át is szerveztük annak érdekében, hogy jobb működési környezetet vagy sajátosságokat kapjon egy másik, nagyobb prioritású alkalmazás. Ez az ABKR nem tudja biztosítani, hogy egy régebben tervezett alkalmazói programot ne kelljen átírni vagy áthangolni működési jellemzőinek javítása érdekében egy olyan időpontnál, amikor az gazdaságosan indokolt (azonnal, később vagy sohasem). A dinamikus kötésnek ezért hangsúlyozott

rangott kell adni az ABKR-ben, mivel nélkül elveszitheti potenciális lehetőségeit akár teljes mértékig is.

Az adatfüggetlenség néhány fontos aspektusa amit a közösen használt adat indukál az alábbi:

- a közösen használt tárolt adat granuláltsága,
- az ABKR által védhető vagy exkluzívan program által tartható minimális kvantuma,
- az a képesség, hogy nevek vagy tulajdonságok alapján azonosíthassuk a kialakított kvantumokat
- redundáns vagy másik adattól függő tárolt adat tartásának, módosulásának automatikus tovártjesztése (beleértve a tárolási hierarchia különböző szintjében lévő példányokat, többszörös példányokat, amiket a látensség hatásának csökkentésére hoztunk létre, háttér kópiákat, indexeket, stb.)
- olyan alkalmazások hozzáadása a jelenlegiekhez azok tudta nélkül, amik nem-konkurrensen használják ugyanazt a tárolt adatot
- konkurrens használatot eredményező nem-ismert alkalmazások hozzáadása
- adathoz való legális hozzáférés automatikus feljegyzése
- a tárolt adat érvényességének és konzisztenciájának automatikus felügyelete
- tárolt adat tartalmának és jellemzőinek módosításáról szóló közlések eredményének tönkretétele és torzítása nélkül.

Alkalmazások széles köre lehet olyan, hogy kezdeti megtervezésükkor a működési követelmények között nem szerepel a közös adathasználat vagy az automatikus felügyelet. Gondosan előre tervezett megosztott adathasználat és biztonsági követelmény deklarációk ele-

gendők lehetnek, hogy a szükséges szintű adatintegritást elérjük a vállalatban belül. Az ABKR-nek azonban automatikus felügyeletre kell képesnek lennie, hogy a szükséges szintű konzisztencia, ellenőrzés/igazolás egyenletek működtethetők legyenek az egyes alkalmazásoknál, vagy adatok bizonyos részein. Hasonlóan a dinamikus kötéshez az automatikus felügyeletnek is magasabb prioritásúnak kell lenni a program specifikációhoz kötött közös használat/konzisztencia felügyelet deklarációkkal szemben, mivel nélkül elveszhetnek potenciális lehetőségei akár teljes mértékben is, és nem lesz jelentős hatása az adatfüggetlenség garantálásában.

Az adatfüggetlenség szoros kapcsolatban van az adatintegritás és az adat biztonság megoldásával az ABKR-en belül. Mindhárom funkció elszigetelt bizonyos mértékben a felhasználót az adatoktól, védik az adatot szándékolt vagy véletlen megérzéstől, behatástól. Ez lényegileg azt jelenti, hogy a programokat védjük más programok hátrányos behatásaitól, amik az integrált adatbázis azonos részein dolgoznak. Az adatfüggetlenség értelmezése körüli zavar rokon az adat biztonság körüli problémákkal. Lehetőség van különböző fajtájú és szintű biztonság deklarálására (ami ABKR képesség) a tárolt adat részeinek kiválasztása, kezelése területen, személyek és végrehajtható funkciók összerendelésére a különböző alkalmazási területeken, másrészt ezen deklarációk helyes használatát felügyelni, megértésére kellő képpen reagálni. Teljes adatvédelmet nem lehet garantálni, de a védekezés apparátusát jól definiált gazdaságossági határig el kell vinni a céltudatos és akaratlan adattartalom megsértés kivédése céljából. Bár az adatvédelem bonyolult kérdéskomplexum (szinte átláthatatlan, el-lentmondásos), mégis lehet specifikálni biztonsági funkciókat és képességeket. Hasonlóképpen a bizton-

sághoz az adatfüggetlenség is ilyen jelenség és a cél adatfüggetlenséget nyújtó funkciókat specifikálni és erre való képességet beépíteni az ABKR struktúrájába.

2.9.2. Az adatfüggetlenség specifikálása.

Az adatfüggetlenség jellemzésére specifikálni kell a rendszerben megengedett manipulációkat. A következőkben megkíséreljük bizonyos területeken az egyes specifikációkat röviden összefoglalni: érvényes műveletek, érvényes transzformációk és dinamikus sajátosságok aspektusából.

a) működési aspektusok: az adatfüggetlenség egy mérőszáma, hogy a tárolt adat tetszőleges aggregátumain vagy a tároló közegen értelmezett művelettipusok milyen mértékben rugalmasak (határosság és célszerűség keretén belül). Ez a képesség a mai ABKR-ek nagy részénél magas fokon rendelkezésre áll és az "eszközfüggetlenség" név alatt szerepel, s a következő művelet fajtáknál jelentkezik markánsan:

- megjelenítés
 - . tabulálás
 - . sor távolság kezelése
 - . lapok formátumozása
 - . grafikus megjelenítés
- módosítás
 - . hozzáadás adathalmaz végénél
 - . felfrissítés adott helynél hosszváltozás nélkül
 - . tetszőleges pozíciónál történő elhelyezés
 - . felfrissítés adott helynél, hosszváltozás mellett

- kiválasztás

- . cim szerint, relativ cim alapján, adatbázis kulcs birotkában
- . elsődleges kulcs alapján
- . adatstruktura halmaz figyelembevételével "tulajdonos" elemre
- előző tagra
- megelőző tagra
- következő tagra
- utolsó tagra
- . kvalifikált elődleges kulcs vagy hozzáférési ut alapján
- . másodlagos kulcs vagy azok tetszőleges függvénye ismeretében
- . tetszőleges adatmezők értéke vagy azok függvénye alapján
- . tetszőleges mezőkön keresztül értelmezett uttal

- irányítás

- . csak a kimenet vagy csak a bemenet felett
- . egyidőben akár a kimenet , akár a bemenet felett
- . konkurrensen mind a bemenet, mint a kimenet felett

b) transzformációs aspektusok: ugyancsak mértéke az adatfüggetlenségnek tárolt adat kollekcióin értelmezett (értelmes és hatásos) transzformációk típusainak flexibilitása. Ez a képesség részben megjelenik a jelenlegi ABKR rendszerekben különböző leképzések formájában, de széles körben nem áll rendelkezésre. Az alábbi lista "Az adatfüggetlenség funkcionális szemlélete" című előadásából való (SIGFIDET, May 1974)

- nincs változás redundanciában

- . cim transzformáció
- . cim és reprezentáció transzformálás (kon-

textus nélküli)

- . cím és reprezentáció transzformálás kontextus függő módon
 - növekedés redundanciában
 - . további indexek létrehozása
 - . további példányok származtatása (átsorrendezett, rendezett)
 - . internális rekord-halmazokba való fragmentálás (projekció)
 - csökkenés redundanciában
 - agglutináció összevonás internális rekord-halmazokba.
- c) Változtatás dinamikájának aspektusai: az adatfüggetlenség mértéke megadható a tárolt adat teljes kollekcióin vagy azok részein végrehajtott transzformációk dinamikájával is. Statikus adatfüggetlenség a fenti korlátokon belül általában rendelkezésre áll a mai ABKR-ekben, bár több kevesebb munkával és megrázkódtatással jár a változtatás. Dinamikus adatfüggetlenség jelenleg egyetlen rendszerben sincs megvalósítva. E területen nagy lehetőségek vannak, de nagyon nehéz specifikálni az adatfüggetlenségi jellemzőket.
- statikus adatfüggetlenség (internális rekord-halmazonként egyetlen forma terjedeleme)
 - dinamikus adatfüggetlenség
 - . rögzített forma terjedeleme határok (új forma terjedelmek hozzáadása internális rekord-halmazokhoz; forma terjedelmek transzformációja exkluzív irányítást kíván az érintett internális rekord halmazokra)
 - . változó forma terjedeleme határok (a forma terjedelmek transzformációja konkurrens lehet a feldolgozással).

2.9.3. Figyelembe vendő tényezők az adatfüggetlenségnél

Ahhoz, hogy teljes, vagy kellően széles skálájú adatfüggetlenség legyen elérhető, azonosítani kell azokat a jellemzőket, amik változhatnak az adatbázis környezetben. A tárolt adatok azon tulajdonságai, amelyek változtatása célszerű lehet a rendszerek teljes életciklusában, az alábbi kategoriákba sorolhatók:

- tárolt adat elhelyezkedése,
- azonosítás vagy megkülönböztetés,
- kapcsolatok reprezentálása,
- értékek reprezentálása.

A tárolt adat elhelyezkedése magába foglalhatja a földrajzi helyet, tároló közeg kötetet, annak részét (sáv, nyom), stb. Az azonosítás önleíró sajátságokat, típuskódokat, leírásokhoz mutató pointereket, stb. foglal magába és a változtatás ezeket érintheti. A relációk reprezentálása tartományokat, indexeket, adatstruktúra-halmazokat, pointer tömböket, láncokat, stb. érint. Az értékrepresentálás skálafaktorokat, precizitást, kódokat, kódolási módokat, stb. jelenthet. Az adatfüggetlenség célja megőrizni egy alkalmazás szemléletének értelmét vagy szemantikáját a fenti négy területen (rendelkezésre állás, kapcsolatok, értékek használata) miközben megengedjük a tárolt adat szintaxisában a változást, eszközök (tároló és hozzáférést adó), adattárolás szervezés, internális mezők szintjén.

A következőkben részletesebben felsorolunk néhány olyan tárolt adat tulajdonságot, amiktől függetleníteni lehet forrásprogramokat. Ez a felsorolás inkább logikai és nem fizikai, nem teljes és nem szigorúan rendezett. A jelenlegi ABKR implementációk adatfüggetlensége kiterjed ezen felsorolás több-kevesebb elemére. Ugyanakkor nyilvánvaló, hogy mindezen jellemzőknek össze kötve kell lenniük a feldolgozó eljárás-

sokkal, mielőtt a tárolt adathoz hozzáférést tudunk adni (statikus vagy dinamikus kötéssel). A tárolt adat jellemzőire a forrásprogramtól függetlenül végezhető változások területei és tárgya:

- mezők közötti kapcsolatok (egy személy csoport valamely egyedének fizetésváltozása hatással van a csoport átlagfizetésére)
- strukturák módosításának milyensége (a szállító megváltozása eredményezhet pointer változtatást vagy szállító név átirást)
- internális mezők maximális mérete, tartománya (fizetés értéke hat számjegyre korlátozódik)
- internális mezők skalárfaktora (pl. méret ezredmilliméterben)
- internális mezők precizitása (pl. érték forintban)
- internális mezők mértékegysége (kp/cm^2)
- internális mezők dimenziója (súly per egységnyi terület)
- internális mezők bekódolása (pl. Pest megye a 4-es számu)
- internális mezők reprezentálása és típusa (standard normált tizedes számu)
- internális mezők kódolása (EBCDIC, ASCII)
- az adatbázis adminisztrátor által adott internális mezőnév
- internális rekord-halmazok neve, ami tartalmazza azon internális mezőket, amelyek megfelelnek az externális mezőknek (hogyan vannak az externális rekord-halmazok szétosztva az internális rekord-halmazokon)
- internális rekord-típusok, amelyekből az internális mezők kivonódnak valamely externális rekordhoz.
- internális rekordok tárolási formátuma és a bele tartozó internális mezők képe (fix hosszúság, szó határokhoz igazított internális mezők speciális betöltő algoritmusoknak megfelelően)

- valamely alkalmazáshoz használt mezőkkel szomszédos mezők biztonsága
- virtuális mezőértékek meghatározásának technikája (algoritmusok)
- konzisztencia fenntartásának technikája (változások átvitel függő mezőkhöz, redundáns értékek előállításának időpontja)
- internális rekordok azon kulcsa, ami entitás azonosítónak felel meg (kulcs helyettesítés)
- internális rekordok visszakeresésének technikája (hozzáférési út, index, táblázat, kulcs transzformáció, végigolvasás)
- bármely internális mező biztonsága, amit érték származtatására használunk fel (egy statisztikai érték általában alacsonyabb biztonságu lehet, mint a származtatásában résztvevő komponensek)
- internális rekord-halmazok adattároló szervezése (pl. indexelt szekvenciális)
- internális rekordok blokkoltsága (blokkonként több szegmens vagy rekord)
- adatokat közösen használó programoknak tárolt adatokkal való kölcsönhatása adatelhelyezési szempontból (internális mezők ugyanazon nyomon vagy szektorban) program változtatás esetén
- több program által használt közös adat szinkronizálása kurrens értékekre program változtatás fenntartásával
- relativ vagy abszolút címen tárolt adat, elismerve azt a szükségszerűséget, hogy mindenfajta, címnek abszolút címre kell átvivődnie
- fizikai rekordok feljegyzési módja és kódja
- internális rekord-halmazokat tároló kötetek
- internális rekordokat tartalmazó kötetek
- kötetek szervezése
- ugyanazon kötetben tárolt másfajta adatok
- szomszédos programok, amelyek ugyanazon köteteket használják és szinkronizálást kívánnak felrakás/

/levétel szempontjából

- tárolt adatot tartalmazó kötetek fizikai és architekturális jellemzői (paritás, szektorhossz és szervezés)
- az a számítógép, amihez az eszköz kapcsolva van (lokális vagy hálózati protokollok és funkciók)
- a számítógép és/vagy eszköz földrajzi elhelyezkedése (kommunikációs út a tárolt adathoz.)

2.10. Adatszótár/adat tartalomjegyzék

Az általános adatbázis modellbeli séma információk és az információs rendszer többi részrendszeréhez tartozó adatbázis tartalmának egy része alkotja azt az ismeret-halmazt, amit "meta adatbázis"-nak nevezhetünk. Ez a meta adatbázis, amit adat szótár/tartalomjegyzéknek nevezünk, tartalmazza:

- az adatbázis sémáiban szereplő adatobjektumok deklarációit,
- az adatbázisbeli objektumok programok részéről történő hivatkozásait,
- az adathozzáférésre vonatkozó statisztikákat,
- biztonsági, adatvédelmi deklarációkat,
- végrehajtás idejü vezérlési strukturákat újraindításhoz vagy hibaállapotból történő feléledéshez (restart/recovery).

Az adatszótár felhasználóinak egyrésze olvassa a benne lévő tartalmat utasítások, követelmények értelmezéséhez döntéseinek támogatása céljából, mások viszont ide rakják le munkájuk során a feldolgozott információ egy részét. Az általános rendszer diagrammon (2. ábra) ezért az adatszótár a transzformációs, feldolgozási utak metszéspontján található, és számos interface körülötte helyezkedik el, amiken keresztül sokrétű felhasználása, illetve folyamatos töltése/módosítása megvalósul. A leglényegesebb interface-eket az adatszótárral a 2,5,14,34,35,36,

37,39,40 és 41-es felületek alkotják. Elvileg lehetséges és törekvés is van arra, hogy ezek az interface-k logikai jellegűek legyenek és egy vagy néhány logikai rekordon keresztül valósuljon meg a sokrétű kapcsolat, amely rekordok, mezők, plexek és rekord-halmazok (ezek metabázisbeli objektumok) hozzáférését, manipulását jelenti.

Az adatszótár körüli interface-kon keresztül manipulált objektumok (rekordok, mezők, plexek és rekord-halmazok) a következő információ féleségeket reprezentálhatják, megemlítve forrásukat, előállítási időpontjukat és az interface számát, amin keresztül áthaladnak:

- felhasználói programok leírásai, amiket a program/eljárás előkészítő alrendszer visz be a szótárba a 7-es interface-n át,
- leképzési struktúrák, melyek a felhasználói programokat és eljárásokat kapcsolják össze az általuk manipulált externális adatbázis objektumokkal,
- externális adatbázis sémabeli objektumok leírásai, amiket az externális adatbázis séma processzor tárol, az 5-ös interfacen át,
- externális és koncepcionális adatobjektumokat kapcsolatba hozó leképző struktúrák, amiket az externális séma processzor tárol az 5-ös interface-n keresztül,
- koncepcionális sémabeli objektumok deklarációi, amiket a koncepcionális adatbázis séma processzor tárolt a 2-es interface-n át,
- az internális adatbázis és a koncepcionális objektum típusok között a 2-es interface-n keresztül megvalósuló leképzési struktúrák, melyeket az internális séma processzor tárol a 14-es interface-n keresztül
- internális adatbázis séma objektum típusok leírásai, amiket az internális adatbázis séma processzor tárol a 14-es interface-n keresztül
- internális adatbázis objektum típusokat és felhasználói programokat kapcsolatba hozó leképző struktú-

- rák, amelyekkel programok processzálják az érintett adatot. Ezt az információt a program fejlesztő alrendszer tárolja a 35-ös interface-n át,
- végrehajtás idejű strukturák, amelyek az újraindítást és a hibából történő feléledést támogatják, másrészt irányítják a közös adathasználatot és segítik a számlázási, passzív felügyeleti funkciók ellátását. Azt az információt a különböző adatbázis transzformációs alrendszerek közlik az adatszótárral a 34, 36 és 38-as interface-n át.
 - adatbázisbeli objektumok és objektum típusok használatára vonatkozó statisztikai adatok, amiket a 34, 36 és 38-as interface-n keresztül kap az adatszótár
 - internális adatbázisbeli objektum típusokat és internális tároló strukturákat összekötő leképzési strukturák. Ezeket az internális adatbázis tervezője hozza létre és a rendszer generálásakor kerülnek bevitelre a 40 és 41-es interface-n keresztül,
 - internális tároló objektum leírásakor, amiket az internális tárolási utility modulok tárolnak
 - internális tároló objektum típusok leírások, amiket az internális tároló rendszer tervezője hoz létre és rendszergeneráláskor tárolódik
 - internális tároló objektumokat externális tároló objektumokkal kapcsolatba hozó leképzések, amiket az internális tároló utility modulok visznek be az adatszótárba
 - externális tároló objektumok leírásai, amiket az externális tároló utility modulok adnak át
 - hozzáférési jogosultság hozzárendelések személyekhez, az adatbázisra vonatkozó (végrehajtható) (specifikált) akciókra nézve. Ez az információ a 2, 5 és 14-es interface-n át jut az adatszótárba a felhatalmazott adminisztrátoroktól,
 - személyekre vonatkozó leírások (kivéve, ha az információs rendszer adatbankjának más részében lesznek

tárolva), amiket felhatalmazott adminisztrátorok adnak meg,

- szöveges (megjegyzés jellegű) állítások a fenti leírásokra és leképzésekre vonatkozóan.

Az adatszótár/adat-tartalomjegyzék az információs rendszer vezérlő adatbázisának része, illetve vele szoros kapcsolatban van. Az, hogy mely részét alkotja vagy milyen módon van vele integrálva, nem tárgya az itt vázolt adatbázis architektúra prototípus leírásának, ameddig az itt specifikált követelményeket kielégítjük. Nyilvánvaló, hogy a programok felismerhetők, kezelhetők legyenek az adatszótáron keresztül. Ugyanilyen integráltságnak (duplikáltságnak) kell létezni a job-kezelő alrendszer adatbázisával is, minthogy az egyes jobokat, azok lépéseit és a folyamatokat ismeri az adatbázis is, és azok adatbázisbeli objektumokkal kapcsolatosak, amiket felfrissítenek és használnak újraindítás vagy feléledés vezérlésekor a megosztott használat, passzív felügyelet és számlázás céljából. Szintén kell, hogy létezzen integráltság a számítókomplexum erőforrásait kezelő alrendszer adatbázisával is, mivel az externális tároló közegeket/egységeket és eszközöket fel kell ismerni. Hasonlóképpen az információsrendszer katalógusa kell hogy ismerje a különböző sémák és rekord-halmazok neveit. Ha az üzenet kezelő alrendszer leírása is rendelkezésre áll "mailboxok" formájában az adatszótár információja alapján lehet teljes adatfolyamat analízist végezni információ forrásoktól információ nyelőkig, a primer rendszer bemenetektől a rendszeren keresztül az adatbázis érintésével valamennyi reális kimenethez az információs rendszerben megvalósuló valamennyi folyamatra.

2.11. Változások és változtatások az adatbázis környezetében

Számos adatbázis és ABKR tervezése arra a kiinduló feltételezésre épült, hogy a kezdeti terv magába foglalja az adatbázis végleges tartalmát és használatának módját.

Sok, a kereskedelembe rendelkezésre álló ABKR ezen elven alapszik és így nem ad lehetőséget rekord formátumok változtatására, olyan tárolt adatok feldolgozására, amit más koncepciójú adatkezeléssel alakítottak ki, vagy saját hatáskörébe tartozó, internális adatok átszervezésére. Azoknak a programoknak, amelyek ezzel az ABKR-el felépített adatokat kívánják használni ismerniük kell a tárolt adatok jellemzőit, szervezését.

A tárolt adataegységek gyakran tartalmaznak olyan csoportokat, amik másik tárolt egységekhez mutatnak és ezeket az alkalmazói programok is manipulálják vagy megjegyzik. Ez a szervezés gyorsítja az adataegységek elérését, az ABKR átbocsátóképességét növeli, de növeli a változtatás költség és intellektuális ráfordítási követelményeit is. A trade-off a változtatásra való képesség és a hozzáférés/átbocsátás sebessége között van, ami akkor dönthető el gazdaságosan, ha hosszú távra pontosan megtudjuk jósolni, hogyan változik az adatbázis rendszer és környezete a tervezett életciklus folyamán.

Az adatfeldolgozás gyakorlatában kialakult egyik alapelv, hogy valamely alkalmazás tényleges fejlődését nem lehet előre pontosan tudni. Segédtevé ehhez az állítás-hoz, hogy egy alkalmazás mindig megváltozik, mielőtt on-line-ná válik. A másik iskolás gyakorlat, ami az előbbi elveken alapszik, azt mondja, hogy az általános ABKR létrehozásának legjobb indoklása, a nyilvánvalóan bekövetkező adatbázis és környezeti változásokhoz való gyors alkalmazkodás. Ez a szemlélet viszont a lokális szub-optimális megoldásokat zárja ki és a változásokkal való megbirkózásnak ad prioritást, aminek túl magas ára lehet a megnövekedett hozzáférési időben és a lecsökkent átbocsátóképességben.

2.11.1. A változások okai

A tárolt adatok változásának elsődleges oka, hogy azok az alkalmazások, amelyek megosztva használják őket fejlődnek, működési tulajdonságaik javulnak, ami az adat tulajdonságainak módosítását kívánja meg. Egyes alkalmazások működtetése meg is szűnhet, miközben más alkalmazások továbbra is használják a vonatkozó adatokat és új alkalmazások beindítása is megtörténhet. A változás sebessége, hatásterülete és az általa okozott effektus kisebb lenne, ha a tárolt adat nem több alkalmazást szolgálna ki egyidejűen, vagy ha nem lenne adatbázisba integrálva. Az integrálás részben hatása, részben oka annak a követelménynek, hogy a tárolt adat konkurrensen, megosztott módon szolgáljon ki független alkalmazói programokat. Emiatt ugyanazon adatnak többféle szemlélete létezhet, melyek internális implementálása közös. Az egyes alkalmazások jóval később kerülhettek kidolgozásra, mint amikor másokat implementáltak. Ez nem változás abban az értelemben, hogy az előbbi adatszemléletet módosítani kellene, hanem továbbiak hozzáadását jelenti, aminek azonban következményei lehetnek. Az ilyen következmények leküzdése miatt szükséges a tárolt adat közös használatának biztosítása olyan programok között, amelyek egymás léteiről nem is tudnak.

A következőkben felsorolunk néhány konkrét okot, ami változást eredményezhet.

- a) A koncepcionális adat fejlődése a vállalatán belül.
 - két vállalatot, vagy részleget egyesítettek, amelyekben az információs rendszerek alapelvei eltérőek voltak
 - két vagy több (vállalatán belüli) adatbázist kombináltak, amikor egy új, eddig ismeretlen kapcsolatot fedeztek fel diszjunktnak gondolt

objektumok között, vagy ismert, de eddig lényegtelennek tartott kapcsolatot kellett definiálni

- Ujabb entitás típusokat találtak a vállalatban belül, vagy újabb koncepcionális rekord típusokat vezettek be az adatbázisba.
- olyan koncepcionális rekord típust találtak, ami több, vagy az előzőktől eltérő koncepcionális mezőtípust kívánt
- valamely koncepcionális mezőtípus közösen használható több koncepcionális rekordhoz is
- valamely koncepcionális mezőt eddigi helyéről másik koncepcionális rekordba viszünk át (Gyakori az is, hogy újabb koncepcionális rekordot deklarálnak, hogy elkerüljék azon programokra való ráhatást, amelyek externális rekordjai azokhoz a koncepcionális rekordokhoz vannak kötve, amelyekhez az új koncepcionális mező hozzáadása fontos.)
- egyszeres mezőtípust többértékűvé kell átalkotítani (azok a programok, amelyek egyértékű mezőként használták eddig, nem szabad, hogy észre vegyék ezt a változást.)

b) Az internális adat fejlődése

- Két olyan adatbázist egyesítettek, amelyek tárolási módja eltérő volt
- Létező internális adatbázist részekre bontottak több olyan szervezeti egységhez (vállalatban belüli osztályok), amelyek alkalmazói igényei izoláltak (ezzel valószínűleg hatásosabb működés biztosítható redundánsan tárolt internális mezők révén az egyes elkülönített adatbázisokban)
- olyan alkalmazás csoportok (valószínűleg konkurrens működésűek), amelyek ugyanazon internális adat részein vagy egészen dolgoznak eltérő konfliktív követelményeket és prioritásokat

kat támasztanak, amit fel kell oldani. Ez a feloldás ugyanazon internális adat különböző részeire eltérő hozzáférési módokat, szegmentálást kívánhat a működési jellemzők teljesíthetőségéhez.

- internális adatmezők tartományát (reprezentálás, hosszúság) ki kell terjeszteni
- közösen használt (konkurrensen) internális mezőknek különböző kapcsolatokkal kell összekötve lenni a különböző alkalmazásokban
- internális mezők vagy csoportok közötti kapcsolatok változnak
- internális mezők, csoportok, rekordok, rekord-halmazok neve megváltozik

c) A rendszer irányítását és optimalizáltságát növelik

- több alkalmazáshoz közösen használt internális adatokat faktorizálják a redundancia csökkentésére
- az internális adatot homogén, egyszerű strukturájú internális rekord-halmazokra bontjuk
- az internális adatot úgy szegmentáljuk, hogy a részek kényelmesebben legyenek felfrissíthetők és változtathatók
- az internális adatot hozzáférési gyakoriság szerint bontjuk (pl. bizonyos adatot on-line nem használunk)
- egymással kapcsolt, vagy kölcsönösen függő adatot egymás közelébe mozgatunk, hogy javítsuk annak biztonságát, integritását és a használat feletti irányítást
- internális adat elhelyezkedését változtatjuk az adatbiztonság növelése céljából
- az internális adatok elhelyezkedését változtatjuk, hogy a külső tárolón rendelkezésre álló helyet jobban kihasználjuk, vagy alkalmazkod-

junk az externális tároló közeg egyedi saját-
ságaihoz

- az internális adat elosztását változtatjuk a kommunikáció költségeinek redukálásához
 - az internális adat tároló szervezése változik az alkalmazások változó követelményei és prioritása miatt (valamely alkalmazást batch feldolgozásról, on-line-ra alakítunk, on-line bemenet késleltetett feldolgozásáról, a tranzakciók azonnali bevitelére térünk át), vagy az internális adat elosztását és mennyiségét változtatjuk. Ez magába foglalhatja másodlagos indexek utólagos kialakítását; átrendezést a hozzáférés gyorsításához, fizikai hozzáadás, vagy eltávolítás végrehajtásának könnyítéséhez; az internális adat újbóli szétosztását forma terjedelmek között, mivel az internális rekord-halmazok különböző részei más-más követelménnyel fellépő, eltérő prioritású, reprezentálású alkalmazásokat szolgál ki
 - az internális adat reprezentálása vagy materializálása változik az alkalmazások változó követelményei és prioritása, vagy a rendszer technikai képességei miatt
 - az internális adat reprezentálása és az adattárolás szervezése változik az externális tároló közeg bizonyos jellemzőinek jobb kihasználása érdekében.
- d) A rendszer konfiguráció változik
- elavult vagy gazdaságtalan működésű komponenseket (software, hardware, firmware) eltávolítunk
 - új technológiák és technikák, rendszerek, komponensek vagy eszközök kerülnek bevezetésre
 - új nemzetközi, nemzeti, ipari, rendszerszintű vagy installációs szabvány lett jóváhagyva, vagy létező szabványt módosítanak.

Az a)-d) pontokban részletezett változások mindegyike olyan, hogy az egyes alkalmazások logikája szempontjából externális hatású és nincs hatása az alkalmazásokra, kivéve, hogy növeli vagy csökkenti a futási időt és egyéb jellemzőket.

- e) Az alkalmazások szintjén fellépő változások
- az alkalmazás az eddigiektől különböző externális rekord-halmazokat kíván munkájához
 - externális rekord-halmazok további externális rekord-halmazok bevonását igényli
 - externális rekord típusok újabb externális mezők beillesztését igénylik
 - egyszerű értékű externális mezőt többszörösre kell változtatni
 - externális mezőtípusok szintaxisa változik (precizitás vagy reprezentáció) mivel az eddigi precizitás nem elegendő az alkalmazásokhoz, vagy a megkívánt precizitás nagyobb, amit egy módosított alkalmazás szolgáltatni képes
 - az externális mezőtípusok szemantikája (értelme) változik
 - externális mezőtípusok, rekordtípusok vagy rekord-halmazok közötti kapcsolatok változnak

Az előbbi változások befolyásolják az alkalmazás belső logikáját vagy algoritmusát, illetve ezek változásának eredményei. Jelenleg nem ismeretesek olyan technikák, amelyek elszigetelik a forrásprogramokat az általuk használt externális rekord-halmazok információ tartalmának változásától. Nem ismeretes miként hat a program belső algoritmusára és eljárására a belső változás és miként függetleníthető attól. Ennél a pontnál az adatfüggetlenség megszűnik és ismert eszközei nem terjednek ki erre a területre.

2.11.2 A változtatások fajtái

Az előbbieken vázolt okok változást eredményeznek, amihez a rendszernek alkalmazkodnia kell oly módon, hogy változtatásokat eszközölünk a vonatkozó sémákban, funkcionális alrendszerekben tároló szervezésbe, stb. A tárolt adat megváltoztatását, a tároló szervezés, adatelhelyezés és reprezentálás módosítását sok egyéb tényező mellett korlátozza a változások dinamikája is. Ennek megfelelően különböző változtatási módokat különböztethetünk meg.

a) Statikus változtatás

A statikus változtatást úgy definiáljuk, mint a tárolt adat definíciójának módosítását a megkívánt tartományban ezzel összefüggésben a tárolt adatnak és minden létező programnak teljes konverzióját oly módon, hogy azok az új leírással komformak legyenek. Ekkor ugyanazon típusú tárolt adat minden előfordulásának ugyanaz a reprezentációja. Például minden internális rekord-halmazban, amit bizonyos alkalmazás-család referálhat egy adott internális rekord típus minden előfordulásának ugyanaz a leírása és minden más rekord típusnak ettől eltérő leíró tulajdonsága is van. Minden programot, ami azt a tárolt adatot referálja, amelynek leíró tulajdonságait változtatjuk meg kell változtatni, hogy az új leírással konzisztensek legyenek. A tárolt adaton addig semmiféle feldolgozást nem lehet eszközölni, amíg az összes tárolt adatot (és valószínűleg a programokat is) át nem konvertáltuk.

b) Dinamikus változtatás

Ezesetben megengedjük, hogy ugyanazon tárolt adat-típusnak konkurrensen létezessen több reprezentációja, amelyek különbözhetnek adattároló szervezésben, reprezentációban, hozzáférési módban, indexelhetőségben, megvalósító algoritmusok tekintetében és sok más egyéb vonatkozásban. Ez az elté-

rés forma terjedelmenként jelentkezhethet és rekord-halmazok határai tekintettel lehet vagy nem. Előfordulhat, hogy gazdaságtalan vagy lehetetlen, hogy a változtatás megtörténte előtt kreált adatot át-másoljuk, vagy nem megengedett a tárolt adat fel-dolgozásának felfüggesztése arra az időre, amíg a változtatást végrehajtjuk. Az ABKR dinamikusán szolgáltatja a tárolt adat egységeket úgy, ahogyan az egyes programok érvényes deklarációkkal össz-hangban várják azokat. Így a tárolt adatot nem kell teljesen átkonvertálni és a programokat sem kell módosítani, hogy a feldolgozás folytatódhasson. A forma terjedelem deklarációk a tárolt adat azon részére vonatkoznak, amelyek követik ezen leíró jellemző halmazokat. Minden forma terjedelemhez, más "sablon" kerül felhasználásra, amint az alkal-mazások által megkívánt módon a tárolt adatot végig-pásztázzuk.

A dinamikus változtatás eredményez:

- fix határokkal rendelkező formaterjedelmeket,
- lehetővé tesz "folytonos konverziót", amely a feldolgozással konkurrensen valósulhat meg.

Ez azt eredményezi, hogy különböző forma terjedelmek keveréke létezhethet együtt egy időintervallumban, vagyis a régebbi internális rekordok a korábbi, az újabbak a javított (vagy új) forma szerint tárolódnak. A folyamatos konverziónál az alkalmazott mecha-nizmus konkurrensen dolgozik, a különböző alkalma-zásokkal és az alkalmazások versenyeznek a változó leírásokkal.

c) Változtatás adott terjedelemben

Fontos különbséget tenni az előbbieken tárgyalt statikus és dinamikus változtatás valamint az ön-

leiró adatok között. Internális mezők értékének reprezentálása állandó lehet a teljes érdeklődési tartományban (pl. internális rekod-halmazban). Ebben az esetben ehhez az internális rekordtípushoz egyetlen leírás tartozik a teljes adatbázisban, amivel statikus adatfüggetlenség biztosítható. Másik lehetőség, hogy a reprezentálás terjedelmenként változik az érdeklődési tartományban. Ekkor a vonatkozó internális rekord vagy az internális mező típushoz minden forma terjedelemben más leírás tartozik, ami dinamikus adatfüggetlenséget nyújt. Harmadsorban az internális mező értékének reprezentációja változhat a mezőtípus minden előfordulására. Ekkor minden előforduláshoz leírás is tartozik (ami lehet a leíráshoz mutató pointer is), azaz önleiró adatunk van.

Fix formátumu internális mezők vagy internális rekordok statikus vagy dinamikus változtatás tárgyai lehetnek. A változó formátumu mezőknek és rekodoknak azonban önleiró típusnak kell lenniük. Az is előfordulhat, hogy az adatobjektumok reprezentálása változik, de a vonatkozó leiró információ a programba ágyazódik be. Ebben az esetben nem tudunk a változáshoz alkalmazkodni, ezért az adatbázis rendszer architektúrája megkívánja, hogy ilyen megoldás ne történhessen, kivéve ha az adatok egyedi (saját) adattárban vannak, amelyek nincsenek az adatbázishoz integrálva.

Célszerű megkülönböztetni a statikus és dinamikus változtatást és strukturákat. Strukturált tárolt adathoz tartozhatnak különböző rekord típusok, amelyeknek saját leírásuk van. A tárolt adatbeli változtatás megengedi a változtatást a leiró jellemzőkben ugyanazon rekord típuson belül is.



d) A dinamikus változtatás gazdaságossága

A dinamikus változtatás több gépidőt kíván a tárolt adat konvertálásakor, amire minden alkalmazáskor sor kerül, szemben a statikus változtatással, ahol a konvertálás egyszer történik meg. Így az előbbi gazdaságtalanabb, de esetenként a változás előtt létrehozott adat átmásolása lehetetlen. Kivánatos lehet esetenként a régebben létrehozott adatot átszervezni gazdaságosabb tároláshoz és az újonnan tárolt adatot (jobb hozzáférés, felfrissítéshez) így fenntartani. Egy idő után azonban az "új adat" is "régivé" válik és inkrementálisan átszerveződik, miközben újabb adatot adunk hozzá az adatbázishoz. Létező programok dolgozhatnak a régi vagy az új adaton, illetve mindkettőn. Ez utóbbi esetben a dinamikus változtatás engedi meg a különböző működtetési tényezők alkalmazását, amik a tárolt adat különböző részeihez alkalmazhatók. Ebben az esetben a dinamikus változtatás a gazdaságosabb.

A dinamikus változtatásra való képesség engedi meg annak a módnak a választását, ami az egyedi szituációhoz a leg gazdaságosabb. Az adatfüggetlenség valódi lényege az, hogy a különböző időpontokban kifejlesztett programoknak lehet különböző szemlélete a tárolt adatról és ezt a szemléletet fenn lehet tartani hosszabb távon is, hogy elejét vegyék a jövőbeli átszervezéseknek internális rekord-halmazokban, míg mások sokkal korábbi adat szemléletet tükröznek, ami nem biztosít hatásos hozzáférést. De a rendszerbe épített adatfüggetlenségi mechanizmus mind az elavult, mind a jövőbeli szemléletet konkurrensen fenntartja alapozva az internális rekord-halmazok konkurrens adattároló szervezésére.

2.11.3 Alkalmazkodás a változásokhoz.

A szokásos programozói technikánál a programozó beépíti az alkalmazói program logikájába és utasításaiba a tároló eszközeire, adattárolás szervezésére és a tárolt adat reprezentálására vonatkozó ismereteit ("optimalizált" programok). A változtatásokhoz úgy alkalmazkodunk, hogy eltávolítjuk ezeket a részeket a forrásprogramból, elkészítjük őket külső formában és leírjuk őket megfelelő paraméterek segítségével. Az ABKR ezeknek a leírásoknak felhasználásával köti össze a programokat az adattal és módosítás után újra kombináljuk őket a programmal.

A tárolt adat csak akkor referálható, ha a program össze van vele kapcsolva. Az összekötés a tárolt adat adatneveinek és tulajdonságainak az externális adatnevekkel és adattulajdonságokkal való szilárd összerendelését jelenti. Az externális nevek és tulajdonságok specifikálása (explicit vagy implicit módon) a tárolt adatot referáló programban történik.

A tárolt adat tulajdonságait a program készítés, és végrehajtás folyamatának különböző pontjainál köthetjük össze egyedileg vagy csoportosan a programhoz. Minél később történik meg a kötés, a program és a tárolt adat között, annál sokrétűbb változáshoz tudunk alkalmazkodni, de annál több interpretív lépést kell végezni. (Jó példája a nagyon késői kötésnek a lebegőpontos számok feldolgozása: a szám skálafaktorát csak a végrehajtáskor kötjük össze a programmal és mindaddig interpretív módon kezeljük akár hardware akár software útján történik a szám kezelése.) Ha a program és a tárolt adat összekötése megtörtént, az adat tulajdonságai nem változtathatók mindaddig, amíg szét nem bontjuk a kötetést majd a változtatott tulajdonságokat újból összekötjük a programmal. Így az adatfüggetlenség a kötés megvalósításának időpontjától függ elsősorban és az "adatfüggetlenítés" csak abban az időtartamban valósulhat meg, amikor hozzárendelés vagy hozzáférés nincs az adattal.

Az alkalmazás lehet a tárolt adat formájában kapcsolt több ponton is, amiből dinamikus vagy statikus adatfüggetlenség származhat vagy az adatfüggetlenség teljes hiánya.

a) Az adatfüggetlenség teljes hiánya

A következőkben példákat adunk forrásnyelvi programokban történő korai kötésekre, amely programozási technikák mindegyike kizárja, hogy a változtatás előbbi formáit alkalmazhassuk. Egy vagy több programot át kell írni (újra kódolni), tesztelni és ismételten a rendszerbe integrálni mielőtt az adat reprezentálás, szervezés, vagy a tárolt adat elhelyezkedése megváltoztatható.

- kompatibilitás biztosítása létező tárolt adat családokkal és programokkal, amelyeket nem lehet (vagy nem célszerű) gazdaságosan megváltoztatni.
- eleve úgy tervezték az alkalmazási családot (implementációs technika, nyelv, stb.) hogy a változtathatóságra nem gondoltak.
- a forrás program írásánál specifikáltak vagy implikáltak az adatleíró tulajdonságokat, a tárolt adat szervezésének módját vagy belefoglalták az algoritmusba a tároló eszköz bizonyos mellék effektusait.

b) Statikus adatfüggetlenség

A következő példák a program feldolgozása vagy végrehajtása idején megvalósuló kötésre vonatkoznak. Ezek a technikák ekvivalensek azzal, hogy a statikus változtatás képességét akarjuk biztosítani. Ha az ABKR-en belül fenntartunk olyan információt, ami az adat felhasználásának helyére utal, akkor a

változáshoz való alkalmazás automatikus lehet. Nyilvánvalóan trade-off tárgya, hogy milyen későn eszközölhetők a változtatások és a program élete során hányszor viselhető el az újbóli összekötés okozta többlet ráfordítás. Hangsúlyozni kell, hogy ha a forrásnyelvi program szöveg tartalmazza a tárolt adat leíró jellemzőit vagy csak explicit állításokat a leíró jellemzőkkel kapcsolatban, akkor már nagyon redukált vagy semmiféle adatfüggetlensége nincs a rendszernek.

- tárgyprogram kompilálás oly módon, hogy be-értjük vagy belemásoljuk a tárolt adatra vonatkozó leíró információt
- tárgy program kapcsolása (linkelése) előre lefordított táblákhoz vagy szubrutinokhoz, amelyek tartalmazzák a leírásokat, illetve ezek a szubrutinok illeszkednek a leírásokhoz.
- adattárak megnyitása, adattárak leíró jellemzőinek összerendelése a tárolt adat leíró jellemzőivel.

c) Dinamikus adatfüggetlenség

A következő példák olyan program / tárolt adat összerendelésre vonatkoznak, ami a hozzáférés időpontjában valósul meg:

- adathalmazok leíró jellemzőinek asszociálása a tárolt adat leíró jellemzőivel minden forma terjedeleme határára
- adattételekhez vagy belőlük felépülő rekordokhoz való dinamikus hozzáférés kihasználva ezen tárolt adat objektumok leíró jellemzőit.

Ezekben az esetekben minden forma terjedeleme határára vagy a tárolt adathoz való minden hozzáférés-

nél adott a dinamikus változtatás lehetősége. Ha a program logika a tárolt adat olyan összességének végig pásztázását kívánja, ami több forma terjedelmében helyezkedik el, akkor a programok a forma terjedelmek jellemzőihez való kötése, illetve bontása a terjedelmek határán történik meg. Ez ekvivalens a folytonos interpretív feldolgozással. A jelenlegi programozási technikák között is szerepelnek olyanok, amelyek a dinamikus változtatás lehetőségét garantálják és mégis jelentősen csökkentik a nagyon késői kötést, azaz az interpretív feldolgozás többlet ráfordítását.

2.11.4. A változáshoz való alkalmazkodás gazdaságossága

Felmerül a kérdés mit nyerünk a változtatási képesség meglétével? A válasz többrétű, mivel a nyereséget önmagában is vizsgálhatjuk, de szembeállíthatjuk az új lehetőségeket a jelenlegi technikákkal és ráfordításokkal is.

Az adatbázis adminisztrátora módosíthatja a reprezentációkat, adattároló szervezéseket és a tárolt adat elhelyezkedését. Így átigazíthatja, áthangolhatja vagy kihangsúlyozhatja a rendszer optimalitását valamely alkalmazás (vagy azok csoportja szempontjából. Ugyancsak lehetséges új berendezések, software és technológia beillesztése az adatfeldolgozó rendszerbe; nem lényeges részek elnyomása, leegyszerűsítése annak érdekében, hogy az alkalmazói programok fejlesztésének költségeit csökkentsük és növeljük a rendszer válaszütemét. Végül lehetséges a költségek szétosztása időben a konvertálási munka szempontjából, mivel a tárolt adat ismételt karbantartását kiküszöbölheti.

Mi a változtatás ára jelenleg? Mivel az eszköz és tároló technológia részletei, az adatszervezés (halmazok) módja vagy az adattároláshoz használt reprezentáció részletei

beágyazódnak az alkalmazói programba, ezért ezen részletek bármelyikének változtatása egy vagy több program átírását, újra tesztelését és ismételt beillesztését kívánja a rendszerbe. Vagyis az adatfüggetlenség hiányáért vagy a rendszer rugalmatlanságáért fizetünk, vagy ha tovább nem tűrhetők ezek a hiányosságok, akkor nagyon bonyolult és valószínűleg csak emberi ráfordítással elvégezhető konverziót kell végrehajtani.

Mi az ára a változáshoz való alkalmazkodó képességnek? A korai adat/program összekapcsolás alacsonyabb szintű adatfüggetlenséget ad, és kevesebb interpretív művelet végzést kíván, míg a későbbi kötés fordított jellegű. Tehát a nagyobb adatfüggetlenségért kisebb átbocsátóképességgel fizetünk. A trade-off a koncentrált ráfordítás (manuálisan végrehajtott konvertálás, aminek költségei egyre nőnek) és az elosztott költség (ami automatikus és költségei csökkenő tendenciájúak) között van.

Az adatbázisok adminisztrálása megkívánja a magasfokú adatfüggetlenséget az ABKR-ben, amivel a rendszer hangolását gazdaságosan elvégezheti. Más szóval fogalmazható ez úgy is, hogy a változásra való képesség, változtatások túlélésének képessége ekvivalens a vállalat azon képességével, hogy túléljen gazdasági, szociális, technikai szituációkat a versenyt jelentő környezetben. Ezen képesség nélkül lemaradhat a versenyben, még ha kipróbált és igaz, de többé már nem elégséges technikát alkalmaz.

2.12 (Humán) résztvevők az ABKR-ben.

Az adatbázis rendszerrel kölcsönhatásba lépő résztvevők három fő csoportba sorolhatók: felhasználók, adminisztrátorok és az adatbázis kezelő (software) rendszer. A rendszerek funkcióinak és a közöttük kialakuló interface-k jobb megértése érdekében célszerű részletesebben megvizsgálni azokat a szerepeket, amiket az egyes résztvevők játszanak a modellezni kívánt vállalat és egymás szempontjából. Az adatbázis adminisztrálás funkcióinak és az adatbázis használatának szétválasztása az adatbázis kezelés specifikálásának egyik lényeges eleme.

2.12.1 A résztvevők szerepe

a) Adat hasznosítás

Az adatbázis használói a vállalaton belül széles skálát fognak át: ügyviteli dolgozók, alkalmazói rendszerek specifikálói, mérnökök rendszer tervezők, programozók, stb. akik procedurális számítógépes nyelveket, lekérdező vagy alkalmazás specifikus nyelveket használnak annak deklarálására, hogy hogyan jelenjen meg az externális adat alkalmazói munkaterületükön, az externális adat hogyan legyen logikailag tárolva az adatbázisban és milyen formában keresik vissza, vagy vizsgálják az externális adatot. Ezek a forrásnyelvi adat deklarációk gyakran az alkalmazói adminisztrátor által előírt formák az illető alkalmazáshoz, miket könyvtárban tart fenn és onnan átmásolhatók a vonatkozó programokhoz. A deklarációknak konzisztenseknek kell lenni azzal a szemlélettel, amit az alkalmazói adminisztrátor definiált egy alkalmazás családhoz, vagy egyedi programhoz. Így az egyes felhasználók léte létrehozza azt a szükségszerűséget, hogy tudjuk adatigényük kielégítéséhez szükséges hozzáférési követelményeket, amelyek externális adat materializálásakor jelentkeznek és a vonatkozó internális adat használatához szükséges felhalmozás megoldását is. A felhasználókat alapvetően az externális adatmodelljük érdekli, ami érdeklődési területük szemlélete és kevésbé érdekli őket a számítógépes modell, ami- ben az internális adat ténylegesen megjelenik.

A felhasználók és különösen az alkalmazási rendszerekanalistái kommunikálnak a vonatkozó alkalmazói adminisztrátorral (aki valószínűleg egy alkalmazási csoport ismerője), vagy ilyen hiá-

nyában a vállalati adminisztrátorral formális és informális csatornákon keresztül. Rögzítik az illető alkalmazás által megkívánt externális adat rendelkezésre bocsátásának módját figyelembe véve az adatbiztonsági, integritási, gazdasági korlátozásokat, amik léteznek a teljes rendszerben és amiket az adminisztrátorok jól ismernek.

b) Adat adminisztrálás

Az adatbázis rendszer kezelése a vállalati adminisztrátor, az adatbázis adminisztrátor és az alkalmazói adminisztrátorok kezében van. Nekik van globális ismeretük, képzettségük és temperamentumuk, hogy szakszerűen, taktikusan és bizalmasan megoldják a felmerülő kérdéseket. Ezek az adminisztrátorok, biztonsági felügyelők vagy adatbázis rendszer fenntartók különböző adatleíró és irányító nyelveket, utility programokat használnak munkájuk során, hogy elveket, eljárásokat, irányítási szabályokat vigyenek be a rendszerbe. Az adminisztrátorok azok a személyek, akiknek érdeklődése főképpen a számítógépes modellre irányul, beleértve a vállalat internális adatát is. A következőkben összefoglalóan "adat adminisztrátor"-nak nevezzük őket, amikor funkcióikat általánosan leírjuk.

Az adatleíró és irányító felelősség magába foglalja az adat feletti felügyeletet és irányítást egyaránt. Az adat adminisztrátorok határozzák meg a fenntartandó, potenciálisan fenntartandó internális adatot, valamint ezek tényleges elhelyezkedését a tárolókon. Ez biztosítja az adatok védhetőségét és a vállalat által megkívánt korlátozások mellett a működési elvárások kielégítését. A definíció formális, precíz leírás az

internális adatra vonatkozóan, reprezentálásra, szervezésre, tárolásra és elérésére. Ugyancsak az adat adminisztrátor hozza létre az adatnak a felhasználók felé rendelkezésre álló szemléletét, valamint a használatához megkívánt felhatalmazási követelményeket, amiket valamelyik specifikus adatszemlélet vagy internális adat használatakor megkívánunk. A közös adathasználat protokolljait, az integritás biztosításához betartandó szabályokat szintén a definiálás hozza létre. A szerviz minőségének, szintjének meghatározása, költség-működési jellemzők közötti trade-off eldöntése, az internális adat átszervezésének, áthangolásának megoldása adott prioritások és követelmények esetén a legoptimálisabb működés biztosításához jól ismert adat adminisztrátori feladat. Az ABKR irányítása az adat felhasználás, felügyeletének, vizsgálatának megoldására, hogy az elérni kívánt célok teljesüléséről, eljárások, irányítási direktívák hatásosságáról ismereteket szerezzünk, a vállalat céljainak teljesüléséről képet kapjunk, szintén az adat adminisztrátorra tartozik. Ezen feladatok ellátása közben az adat adminisztrátort számos tényező korlátozza. Ezek közül az internális adatban rejlő tényleges képesség, a törvényes előírások, működési megkötöttségek és a real-time működéséhez megkívánt válaszidő előírások a legjelentősebbek.

Fontos különbséget tenni felügyelet, irányítás és tulajdonosság között. Minden internális adatot a vállalat birtokol, de bizonyos adatrészek használata, felügyelete lehet máshoz delegálva. Ebben az értelemben bizonyos osztályok vagy alkalmazások "tulajdonsága" lehet az internális adat egy része és az alkalmazás orientált köve-

telményeket, karbantartási szabályokat, felhasználásra vonatkozó előírásokat javasolhat a tulajdonos, másrészt az internális adat tárolási, karbantartási költségei vele fizettethetők. Az internális adatot megőrzésre, fenntartásra és irányításra át lehet adni az adat adminisztrátornak, aki ezeket a funkciókat gyakorolja anélkül, hogy magának az adatnak a használatára felhatalmazták volna.

Az adminisztrálás a centralizálás és az adatbázis rendszer feletti irányítás gyakorlására létrehozott ügyrend. Magába foglalja, hogy személyek egy csoportja felelős az adatbázis rendszer működéséért és a kitűzött stratégiák teljesüléséért. Ezeken túl ide tartozik néhány speciális feladatkör is, nevezetesen az információs rendszer analisták, adatstruktúra és adattároló szervezés tervezők, biztonsági intézkedések specialistái, rendszer fejlesztő szakemberek, stb. munkája. Nagy, komplex, erősen megosztott használatra tervezett adatbázisnál ezeket a feladatokat nem tudja egy személy ellátni, de egyszerűbb esetekben áttekinthető alkalmazási környezetben egyetlen személyhez delegáló valamennyi előbbi feladat.

Az adat adminisztrátor a számítógépet használja eszközeinek létrehozásához és azok működtetéséhez. Ilyen eszköznek számítanak a rendszerbeli adat és információ áramlás analízisére szolgáló eszközök, adatköri kapcsolatok analízisére, adattároló szervezésre, újraszervezésre szolgáló utility programok, integritás és konzisztencia igazoló és helyreállító eszközök. Az adott számítógépes környezetben meglévő adatfeldolgozó hardware és software rendszerek automatikus működési jellemzőitől függően (ahogy azok önsza-

bályozóan, szituációtól függően automatikusan működésbe lépnek) az adat adminisztrátor fel szabadul az előbbi eszközök működtetésének terhe alól. Az információ folyam analízis és adattároló szervezés analízis eszközeinek használata révén az adat adminisztrátor optimálisabb irányelveket alakíthat ki. Amennyiben a segítő software eszközök képesek alkalmazói rendszer követelményeknek adatleíró jellemzőkké való átképzésre, amelyek az adat adminisztrálás, irányítási, áttekintés tárgyai lettek volna, akkor az adat adminisztrátor ezen munka nagy részétől is tehermentesíthető. Minél átfogóbb és automatikus ezen eszközök funkciója az ABKR-ben, annál jobban szabadul fel az adat adminisztrátor a munka részleteivel való foglalkozástól, viszont az egyre bonyolultabb automatizmus miatt az adminisztrálás funkciója csökken ugyan, de sokkal nehezebbé válik.

Az, hogy az adat adminisztrátor gyakorol-e helyi, manuális hangolást az internális adat részein, vagy teljesen elfogadja az automatikus eszközök működésével kialakuló strukturákat, az emberi és gépi munka ráfordításai közötti arányoktól függ.

c) Az irányítás centralizálása

A centralizálás nem jelenti feltétel nélküli hegemonia létrehozását minden tevékenység felett. Ha két alkalmazás csoport nem használ közös tárolt adatrészeket és nincs erőforrás konfliktusuk sem, akkor vitatható, hogy szükséges-e egyetlen fölérendelt hatóság felettük. A megoldás módja vezetői stílus kérdése. Ugyiszintén ha bizonyos adat fenntartásához személyes felelősség is társul, de azt nem használja más fel-

használó, továbbá törvényességi és gazdaságossági szempontból lényegtelen a vállalat munkájához a felette való irányítás, akkor nem szükséges, hogy ezt az adatot a többi internális adathoz integráljuk és nem vetjük alá szigorú felügyeletnek. Ugyanakkor jól ismert tény, hogy ami kezdetben helyi és egyéni adat, az idő során lényegessé és közösen igényeltté válhat.

Az a képesség látszik elengedhetetlenül szükségesnek, hogy a centralizálást gazdaságosan hajtsuk végre akár valamely alkalmazás vagy tárolt adathalmaz létrehozása után is. Ha a felügyelet helyi megoldásának mechanizmusa konzisztens az irányítás globális centralizálásának megoldásával, akkor éppolyan hatásos lehet az irányítás centralizálása az adatbázis elválasztott részén egy alárendelt adminisztrátorhoz, mint a teljes rendszerben. A helyi adminisztrátor ugyanúgy tevékenykedhet, mintha globális irányító lenne. Képesnek és informálnak kell lennie, hogy alkalmazások családját kezelje, megszervezze, gazdaságosságát, használatát, adatainak tárolását biztosítsa. Az adat adminisztrátor nem adhatja át vagy hanyagolhatja el az irányítást az adatbázis részei felett az alkalmazások ellenében. Nem mondhat le azok irányításáról és semmiképpen sem engedheti meg, hogy alkalmazások forráskódjába, deklarációiba épüljön be a tárolt adat valamelyik leíró jellemzője, mivel ekkor elveszne a tárolt adat feletti irányítás centralizálásának lehetősége, ami a vállalat későbbi gazdaságossági döntéseit (adatszervezésre, vonatkozóan) akadályozná. Ha az irányítást helyileg már centralizáltuk és ezután válik szükségessé a tárolt adat közös használata, akkor az irányítás globálisan is centralizálható anél-

kül, hogy létező programokra nem kívánt mellékhatások lépne fel, vagy az adatot át kellene szervezni. Ugyanígy célszerű lehet a decentralizálás lehetőségét biztosítani, ami után nem zárunk ki az újbóli centralizálást sem egy későbbi időpontban.

Elválasztva az adatbázis leíró és irányító funkciókat a (forrás) programoktól lehetővé válik irányítási stratégiák létrehozása, továbbfejlesztése, végrehajtása folyamatosan és általában a forrásprogramok változtatásának tetemes költsége nélkül, ami esetenként a változtatások kivitelezhetetlenségét eredményezi. A hosszabb távú stratégiák feletti irányítás centralizálása a vállalatnál biztosítja, hogy azok a vállalat egészének követelményeit tükrözik és nem egyedi, rövidtávú érdekeket. Az alárendelt csoportok valószínűleg nem ismerik a teljes vállalat adat szervezésének gazdaságosságát, különösen azokat a szektorokat nem, amelyek ismeretére vagy használatára felhatalmazásuk sincsen. A tárolt adatok kezeléséhez, optimalizálásához szükséges műszaki szakismeretek skálája széles, aminek el-sajátítása nagy ráfordítást (képzés) kíván, amit nagyon sok embernek átadni nagyon költséges lehet. A centralizált adminisztrálás ezenkívül kiegyensúlyozhatja a teljes vállalat el-lentmondásos szükségleteit és prioritásait, továbbá legnagyobb visszatérülést biztosítja az adatfeldolgozásba fektetett beruházásoknak.

d) Az adatbázis kezelő rendszer

Az adatbázis kezelő rendszer hardware, software és firmware kombinációja, amely támogatja a felhasználók és az adminisztrátorok interface-ét az adatbázisban. A vezetés és felhasználók felé

információt bocsát át, mivel informálva van a vállalat internális adatkollekcióiról, azok elhelyezkedéséről, használatuk szabályairól. Ismeri a vállalat követelményeit, prioritásait, gazdasági stratégiáit, a "tabu"-jait is, amiket végre is hajt. Konvertálja az adat adminisztrátorok deklarációit (leírások és szabályok) rendszerorientált táblákba. Externális adatkereséseket dolgoz fel, elfogad externális felhasználóktól származó adatot és fenntartja az internális adatokat, amikből a deklarációkkal konzisztens módon az externális adat létrehozható a felhasználó által megkívánt módon. Eközben figyelembe veszi, a felhasználókra, az internális adatra és más, az adat adminisztrátorokra vonatkozó korlátozásokokat (biztonság és integritás). Az ABKR a megfelelő helyen és időben elvégzi az összes automatikus műveletet, amiket az integritás fenntartása az adat használat feljegyzésére vonatkozóan az adat adminisztrátorok deklarálnak. Az AEKR függ az aktuális operációs rendszertől, ami irányítja a terminálokat és egyéb bemeneti/kimeneti eszközöket, hozzárendelését és szinkronizálását, a megfelelő prioritások és erőforrások hozzárendelését, ami a számítógépes műveletvégzés környezetének létrehozását jelenti.

2.12.2 A résztvevők jellemzése

A résztvevők általában ember/gép interface-knél teljesítik feladataikat és szerepüktől függően vonásaik, hatáskörük is eltérő. A humán résztvevőkre adható meg egyértelműen ilyen jellemzés

a) Adminisztrátorok

1. Vállalati adminisztrátor

A vállalat koncepcionális sémája és adatbázisa felett gyakorol elsődleges irányítást. Mivel ismeri a vállalat szervezetét, műveleteit, információ tartalmát és strukturáját, ő határozza meg, deklarálja a koncepcionális modellt. Irányelveket ad az internális adat használatára és elérésének szervezésére, amiből a koncepcionális adat materializálható. A koncepcionális adat strukturáját, biztonságát és integritását szabja meg, korlátozva ily módon, hogy ki által, hogyan és melyik koncepcionális adat manipulálható vagy érhető el. Az előbbiek valójában a koncepcionális séma definiálását jelentik.

2. Adatbázis adminisztrátor

Az internális adatbázis és az adatbázis fizikai megtestesítésének felelőse. Fő funkciója az internális adatok gondozása, azaz biztosítja épségét, könnyű helyettesíthetőségét és hatásos szolgáltatást ad internális adat reprezentációk választásához, indexelési mechanizmusok képzéséhez, egyéb eszközöket adattároló szervezéséhez, externális tároló eszközök kiválasztásához, elhelyezési stratégiák meghatározásához. Felelős az internális adattömeg redundanciájáért, áthelyezéséért/áthangolásáért az internális adatbázis egyes részeiben, ha

az indokolttá válik. Az adatbázis adminisztrátor definiálja az internális sémát, valamint a koncepcionális modellnek internális modellbe való leképezését biztosító transzformációt.

3. Helyszini operátorok

A rendszert általában, rutinszerűen a számítóközpont operátorai működtetik, akiknek funkciója, hogy felügyelnek az adatfeldolgozó berendezések és programok normál működésére. Következésképpen ők válaszolnak az ABKR által generált reportokra, üzenetekre (gyakran automatikusan és mechanikusan) az adatbázis rendszer percről-percre történő működtetésével. Ők rakják fel/veszik le a szalag és diszk köteteket és gondozzák az adat generáló és nyelő eszközöket.

4. Alkalmazói adminisztrátor

Az alkalmazói adminisztrátorok könnyítik meg, hogy egy alkalmazás hozzáférjen externális adatának azon részeihez, ami megfelelő az illető alkalmazás családdhoz és ők készítik el a deklarációkat, hogy az alkalmazás externális adata olyan formában jelenjen meg az alkalmazói programok munkatüreltén, ahogyan a leghatékonyabb munkájukhoz. Rendszerint egy-egy alkalmazás család működéséért felelősek (gyártás, elszámolás, személyzeti, stb.), amelyek tartalmazzak algoritmus részeket és igényelnek adatbázis funkciókat. Az externális sémákat az alkalmazói adminisztrátorok definiálják.

b) Felhasználók

1. Rendszer programozók

A teljes információs rendszerben (beleértve magát az ABKR-t is) a rendszer programozók felelősek a tervezett módosításokért, és a rendszer képességeinek kiterjesztéséért adatbázis eljárások kialakításával, valamint a rendszer paraméterek illesztésével. Egyszerűbb rendszerekben a helyszini operátorokat és az adatbázis adminisztrátort is ők segítik a hibaállapotból történő fellesztéskor, újraindításkor abnormális viselkedés után. Autaomatikus rendszerekben ezt a rendszer által fenntartott katalógusok (journal-ok), feljegyzések (log-ok) alapján rendszerprogramozói segítség nélkül végzik a utility programok. Máskor a központi tár és a háttér bit-manipuláló segédletével manuálisan hajtják végre az újraindítást.

2. Alkalmazói programozó

A különböző feladatokat ellátó programok írását végzik, amelyek használják az externális adatot, ami internális adatból hozható létre. Általában jól definiált feladatok ellátására írják az alkalmazói programokat parametikus felhasználók részére, akik kevés ismerettel rendelkeznek a számítógépes technikákat illetően, csupán behívják az előre definiált programokat bizonyos paraméterek és bemenő értékek deklarálásával.

3. Vég-felhasználók

Általánosan így nevezzük azokat a személyeket, akikről elvárható az információ hasznosítás (ami az adatbázis internális adataiból származtatható) a számítógépes rendszerre vonatkozó átfogó képzés nélkül, vagy alkalmazói programok írására való kiképzés nélkül is. Két fő típusuk létezik, akik:

- előre nem tervezett, nem strukturált érdeklődést mutatnak az információ iránt (vizsgálódást folytatók)
- tervezett, rutinszerű, ismétlődő, strukturált érdeklődők az információ iránt.

A nem strukturált folyamatokat kezdeményezőktől azt várja a rendszer, hogy az adat-szótár szolgáltatásait is igénybe veszik annak megismerésére, hogy egyáltalán mi áll rendelkezésre, mi segítheti őket feltételezéseik vizsgálatában. A másik csoport csak az előre definiált információ részt biztosítja és nem érdeklődik más információ iránt.

A strukturált folyamatok felhasználói négy fő csoportba sorolhatók:

- lekérdezés specifikálók
- felfrissítési akció specifikálók
- report írást specifikálók
- parametikus felhasználók

A parametikus felhasználók kivételével ezek a felhasználók nagyon magasszintű interface nyelvet használnak, amely lehet procedurális vagy deklaratív karakterű.

A nem strukturált folyamatok kezdeményezői előre nehezen megjósolható internális adattömegnövekedést tapasztálnak végig és nem várható, hogy permanens, vagy erősen formátumozott kimenetre töreksszenek. A lekérdezési specifikáló várhatóan előre látható adathalmazt néz végig és kis volumenű kimenetre számít. Ezzel szemben a report specifikáló erősen formátumozott alakban nagy adattömeget vár. Felügyeletet nem igénylő felfrisztési akciók általában korlátozottak képességeikben, mivel ezek képesek az adatbázis integritásának megsértésére. A magasabb szinten automatizált rendszerekben a felfrisztési akciók feljegyzésre és diagnosztálásra kerülnek, ami megakadályozza a szándékos vagy véletlen integritás megsértést. Ekkor lehetséges, hogy az érvényesnek (elfogadhatónak) ítélt módosítások tovább terjeszthetők legyenek a redundáns és függőségi viszonyban lévő adatokhoz az információ részek közötti konzisztencia fenntartása érdekében.

A parametikus felhasználó előre definiált eljárásokat hív be, amelyeket a lekérdezés, felfrisztés vagy report specifikáló irt meg saját nyelvén vagy alkalmazói program formájában került kialakításra. Ezek az eljárások jól definiált bemenő paramétereket és bizonyos bemeneti értékeket várnak a felhasználóktól. Ez az interface mind természete, képessége és nyelve szempontjából különleges és a parametikus felhasználók csoportjai számára kerül kialakítására. Alapvetően azok a funkciók határozzák meg ezt az interface-t, amiknek ellátását várjuk az illető parametikus felhasználótól.

III. AZ ADATBÁZIS RENDSZER MŰKÖDÉSE AZ INFORMÁCIÓS RENDSZER KÖRNYEZETÉBEN

Az előző fejezetben leírtuk a javasolt adatbázis rendszer általános architektúráját, működését és definiáltuk azokat az elveket, fogalmakat, amelyek lényegesek az implementációs megfontolásokhoz. A következőkben a működés olyan vonatkozásait tárgyaljuk, amelyek a modell bizonyos szintjei között kapcsolatokat és az egyes szintek kölcsönhatásba lépő elemeit írják le. Ez alapot ad ahhoz, hogy az egyes interface-kat jól körülhatárolhassuk és leírásukat megkönnyítsük.

A következőkben a modell működését három fokozatban mutatjuk be, ami megfelel annak a három fő fázisnak, amit egy információs rendszer adatbázisa installálásánál követünk:

- sémák kialakítása,
- program fejlesztés,
- program végrehajtás.

Az egyes feladat csoportok több lépést tartalmaznak, amit ábrákkal illusztrálunk. Ezek fokozatosan "kitöltik" a teljes adatbázis rendszer egyes blokkjai és végül az általános (2. ábra) adatbázis rendszerhez jutunk, amit most a 3.0. ábra tartalmaz. Az ábrán a körülkerített rész jelenti és tartalmazza a hardware specifikus interface-okat, amikkel a jelen ABKR modell nem foglalkozik. Ezek szabványosításra kevésbé alkalmasak és az itteni interface-k lényeges változásokon mennek át a tároló eszközök és technikák gyors fejlődése következtében. A következő leírás ezek vizsgálatával nem foglalkozik, de az ábrákon feltüntetjük ezeket is, hogy lássuk a velük való kapcsolatokat és folytonosságot kapjunk az érdeklődési területek között.

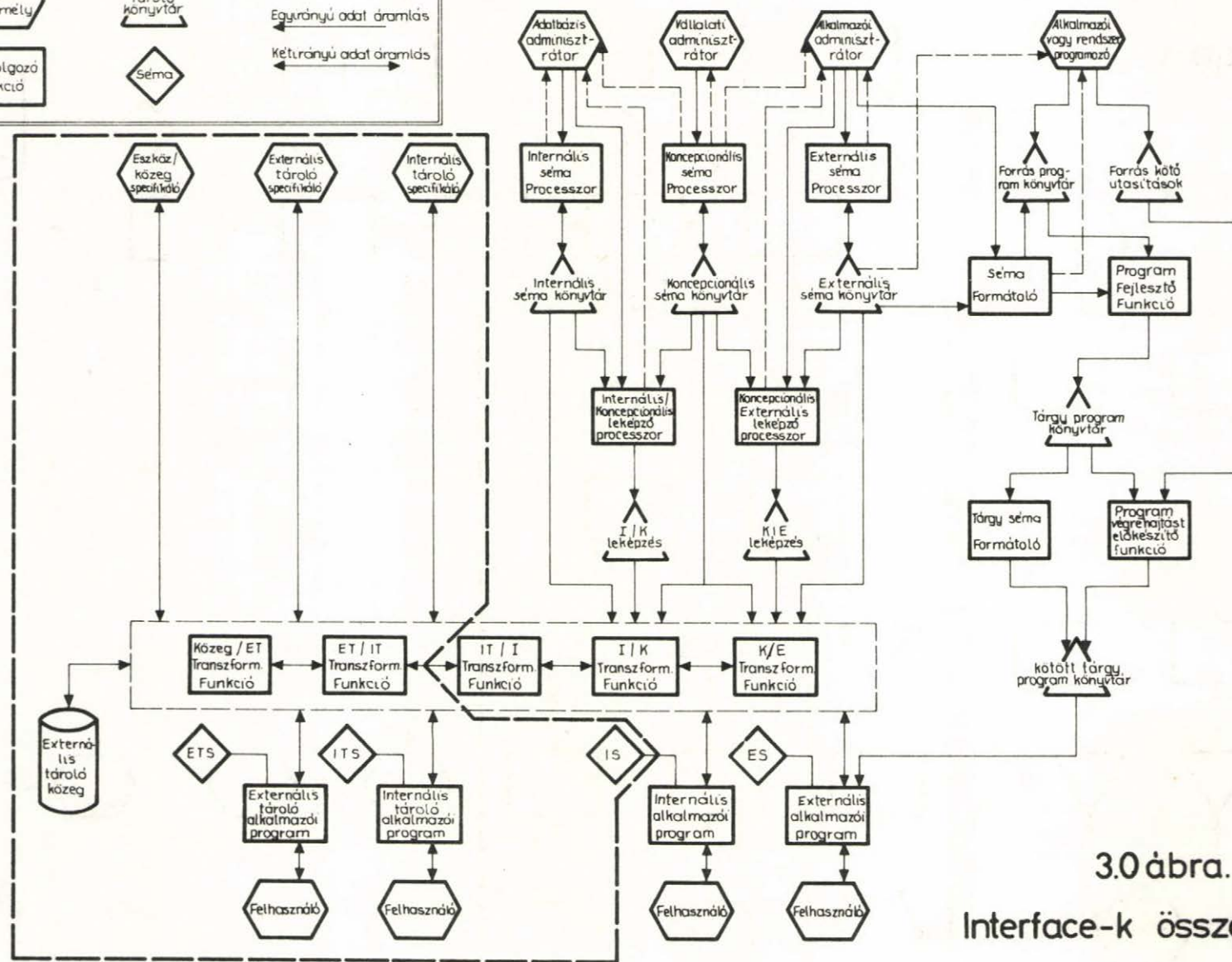
3.1 Sémák kialakítása

3.1.1 A koncepcionális séma előkészítése (3.1 ábra)

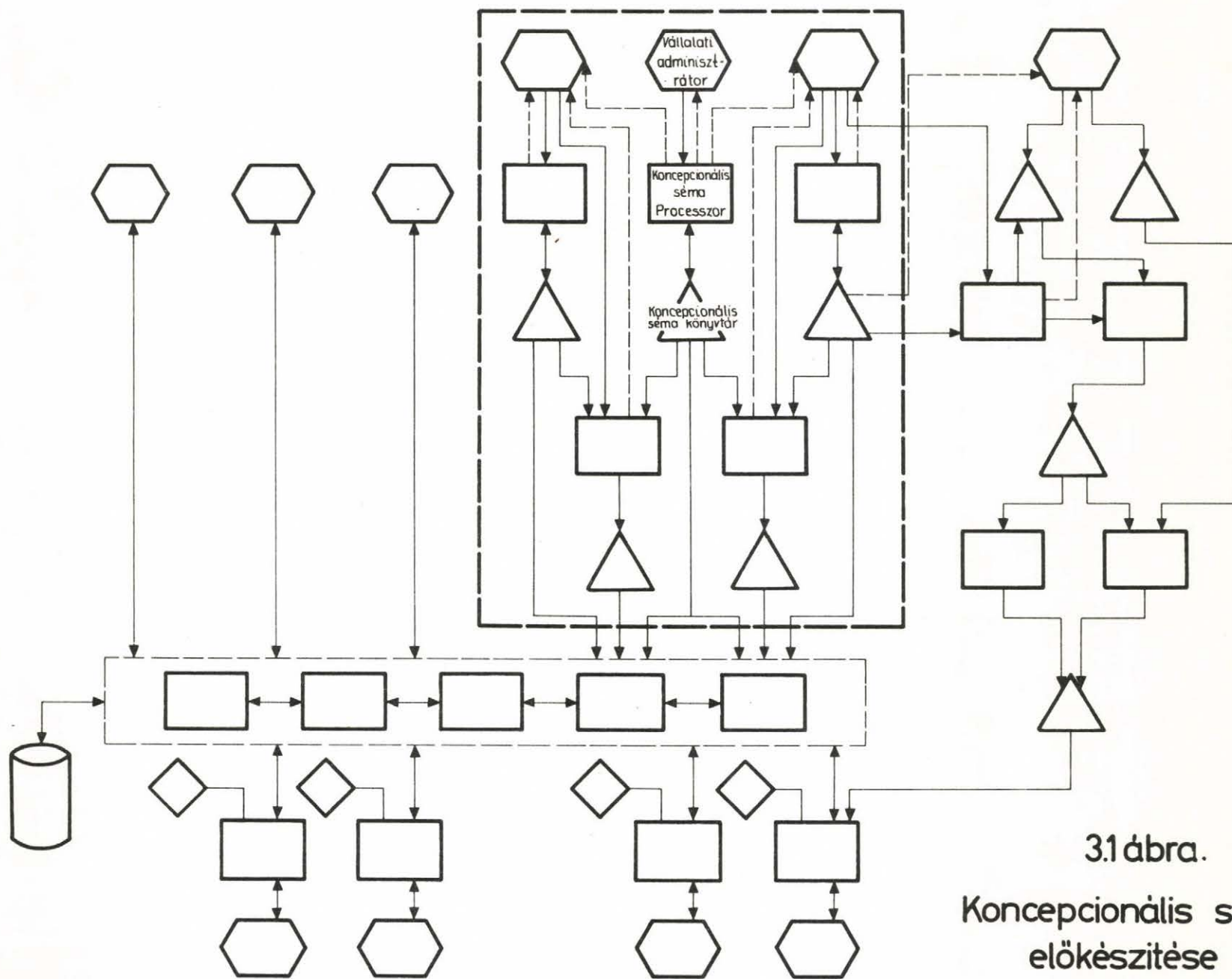
Mielőtt egy adatbázis kifejlesztését elkezdenénk, szükséges annak a környezetnek alapos megismerése, amit az szolgálni fog. Következésképpen fontos kezdeti feladat a vállalatban belüli információ igény jellemzése, feltárása és szintézise. Meg kell határozni, milyen információ áramlik keresztül a vállalatban és hogyan hasznosítják azt. A vállalati adminisztrátor legfontosabb szerepe az, hogy ezt a rendszer szintézis funkciót ellássa az adatbázis használó sokféle alkalmazás kontextusában.

A vállalati adminisztrátor játszik központi szerepet a szervezetben belüli információ felhasználás (és előállítás) meghatározásában. Ekkor kell meghatározni, hogy milyen információ kezeléssel kell gondoskodni, milyen biztonságot, integritást és rendelkezésre állást kell garantálni az illető információ objektumhoz és információ darabokhoz. Ezen funkció ellátása közben a vállalati adminisztrátor leírja az

JELMAGYARÁZAT



3.0 ábra.
Interface-k összefoglalása



31. ábra.
Konceptcionális séma
előkészítése

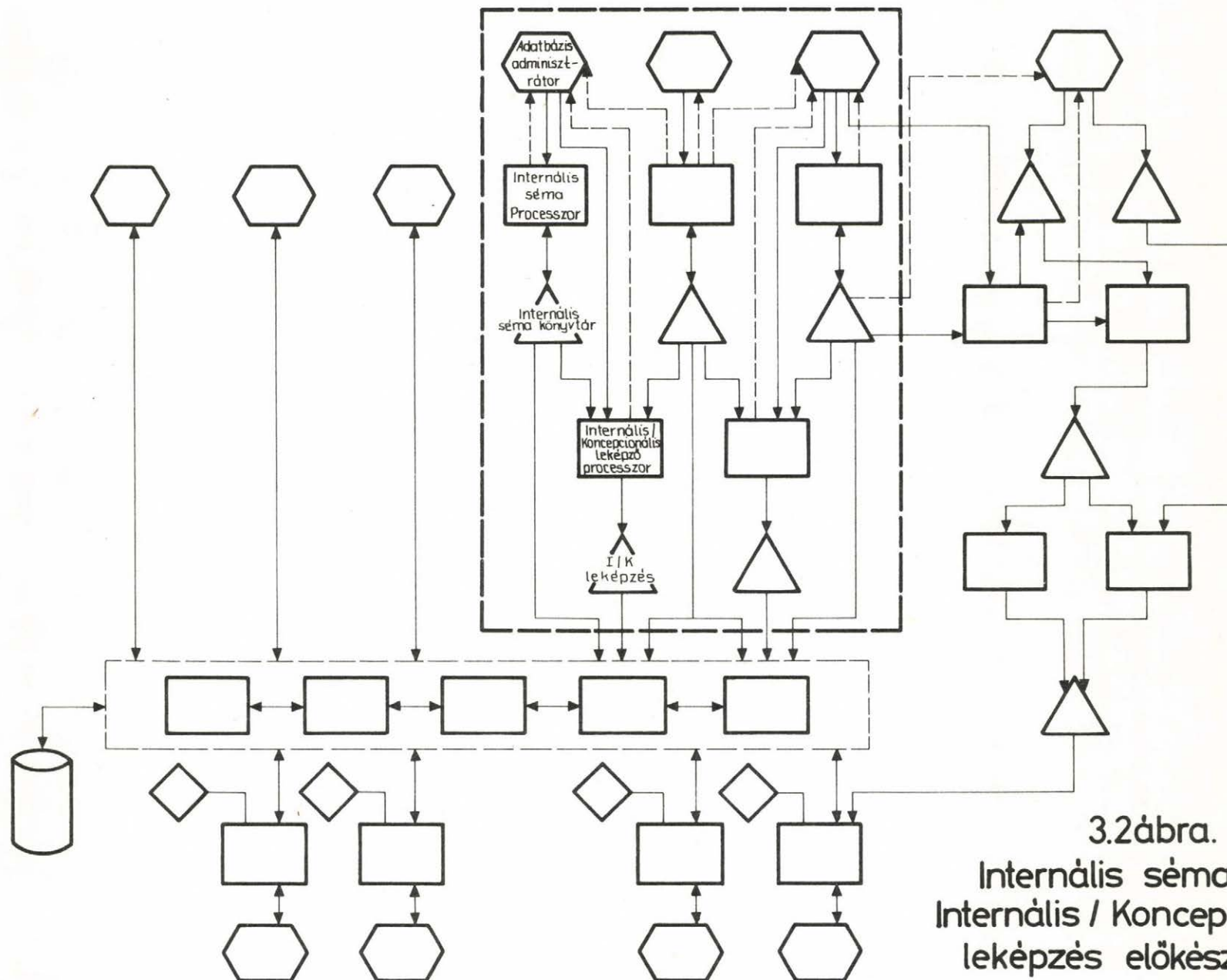
adat szótár/adat tartalomejegyzék egy részét. Így az információ analízis és rendszer szintézis végterméke úgy tekinthető, mint az adat szótár/tartalomejegyzék "első szintű" leírása.

Ha a vállalati adminisztrátor megértette a szervezet információ igényét és definiálta az információ használatát, áramlását és hozzáférhetőségét a rendszeren belül, akkor a koncepcionális séma leírható valamely formális "leíró" nyelven. Ez szolgál a következőkben a vállalat információs modelljeként és belőle származtathatók az adatbázis rendszerrel szembeni követelmények. Ebben a szerepben a vállalati adminisztrátor tölti be az interface-k funkcióját a szervezeten belüli műveletek között, összegyűjtve minden tényt és információt, amik végül is az adatbázis adminisztrátorhoz lesznek átadva elősegítendő a működési problémák megoldását. Ugyancsak rendelkezésre áll ez az ismeret az alkalmazási adminisztrátoroknak is, hogy az alkalmazói programok tervezését és implementálását segítse.

A kialakított koncepcionális sémát átadjuk a koncepcionális séma processzornak, ami bekódolja számítógéppel olvasható formára. Közben szintaktikai és logikai konzisztencia ellenőrzéseket végez az információs leírásokon és kapcsolatokon, hogy felderítse az esetleges inkonzisztenciákat a vállalati adminisztrátor által leírt koncepcionális szemléletben és modellben. A koncepcionális séma processzor a koncepcionális séma leírását külön könyvtárban tárolhatja.

3.1.2 Az internális séma előkészítése (3.2 ábra)

Miután a koncepcionális sémát előkészítettük, feldolgoztuk és katalogizáltuk, az adatbázis adminisztrátor felelőssége, hogy specifikáljon egy lehetséges internális modellt (vagyis a fizikai adatbázist)



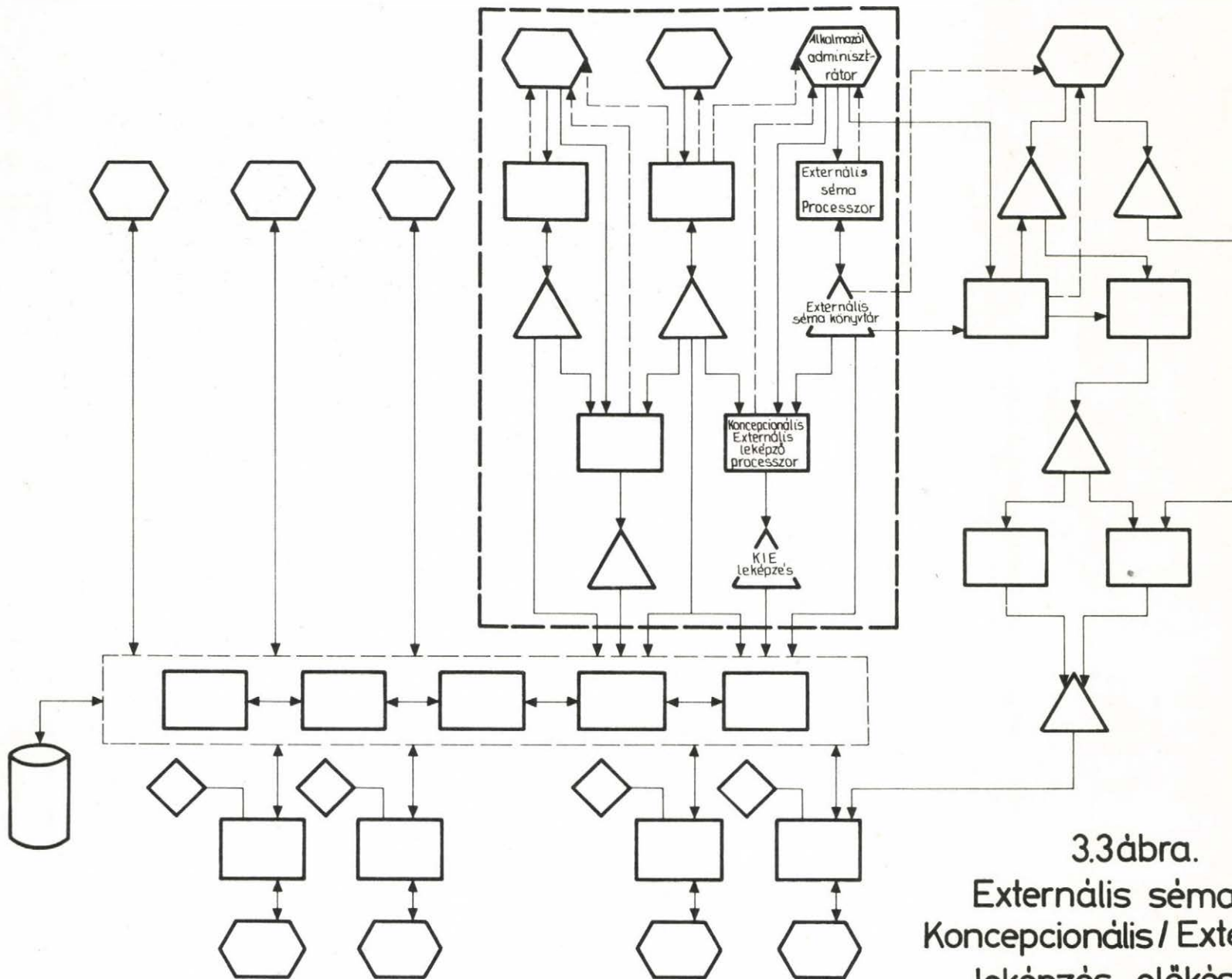
3.2ábra.
Internális séma és
Internális / Konceptcionális
leképezés előkészítése

arra az információra, amit a koncepcionális sémában leírtunk. Ezen feladat ellátásához az adatbázis adminisztrátor meghatározza a használatra vonatkozó követelményeket az adattartalomra és a hozzáférési módra vonatkozóan, mivel az alkalmazások ezért a két "erőforrásért" harcolnak. Az internális modell tükröz bizonyos tényeket arról a környezetről, amelyben az adatbázisnak működnie kell. Néhány dolog ami befolyásolja az internális modell strukturáját a következő: a teljes rendszer működése; a válasz idő; várható rendszer terhelés; konkurrens használat változó környezetben az előbbi jellemzők szempontjaiból. Erősen befolyásoló tényező, ha szükségessé válik a programírás internális szinten is és az internális modellben a koncepcionális sémában specifikált biztonsági és integritási szabályokkal konzisztens biztonsági/integritási mechanizmust kell működtetni. Az internális modell implementálása erősen kötött a hozzáférési módszerekhez és technikákhoz, objektumok közötti kapcsolatok reprezentálásához.

Ezeknek a kérdéseknek a feloldása adja az adatbázis adminisztrátor számára azokat az alkotórészeket az internális séma specifikálásához, amiket aztán az internális séma processzor konvertál olyan formára, amit az ABKR interpretálni tud. Az internális séma az internális séma könyvtárban tárolhatja az internális séma processzor.

3.1.3 Externális sémák előkészítése (3.3 ábra)

A különböző alkalmazói programok, amik a vállalat adatbázisát használják az alkalmazói programok iránnyitása alatt vannak. Az alkalmazói adminisztrátorok a vállalati adminisztrátorral közösen határozzák meg az egyes alkalmazásokhoz lényeges objektumokat. Minden kialakított externális sémát egy vagy több alkal-



3.3.ábra.
Externális séma és
Konceptcionális / Externális
leképzés előkészítése

mazói program használhat. Az externális sémának valamely alkalmazáshoz illeszkedő tervezésénél az alkalmazói adminisztrátor konzultál a többi alkalmazói program externális sémáinak kialakítóival annak meghatározására, hogy milyen adat áll már rendelkezésre internális szinten.

Az alkalmazói adminisztrátor felelőssége, hogy interface-t képezzen az alkalmazói programozók felé (vagy rendszer programozók felé) és segítse őket a programok készítésénél annak az externális sémának a felhasználásával, amit részükre kialakított. Valamely externális séma kialakításánál az alkalmazói adminisztrátor használhat programozói nyelvtől független leíró nyelvet, vagy valamelyik programozói nyelvhez illeszkedő nyelvet. Az alkalmazói programozók számára úgy tesszük lehetővé programjuk adatleíró részének definiálását, hogy az externális sémát átadjuk nekik, többnyire indirekt módon egy externális séma formátumozón keresztül, ami kényelmes használatot, áttekintést tesz lehetővé. Fontos megjegyezni, hogy a különböző alkalmazásoknak lehet eltérő externális szemlélete ugyanazon koncepcionális és internális adatbázisról. Ezen eltérő szemléleti módok olyan externális sémák létrehozását eredményezik, amik relevánsak az illető alkalmazásokhoz.

Az externális séma processzor az előbbi processzorokhoz hasonlóan a kialakított externális sémákat séma könyvtárban tárolhatja, ahonnan rendelkezésre bocsátja az egyes alkalmazói programok számára.

3.1.4 Leképezések a koncepcionális, internális és externális sémák között.

A sémák közötti megfelelés biztosítására az adatbázis adminisztrátora leképezéseket specifikál. Ezeket a leképző függvényeket az adatbázis rendszer használja,

amikor logikai szintű adatkereséseket értelmez és lefordít belső szintű hozzáférési műveletekké. Az alkalmazói adminisztrátorok az externális és a koncepcionális sémák közötti leképezéseket határozzák meg és ezzel valósul meg az alkalmazói programokból származó externális orientációjú adatkérések lefordítása (az adatbázis rendszer részéről) koncepcionális orientációjú kérésé.

A leképezések kölcsönös megfelelést specifikálnak különböző sémákban szereplő objektumok között: meghatározzák az objektum nevek összefüggéseit, szabályokat adnak az információ formátumának változásához (beleértve az adattípus konverziót is) rész halmazokat képeznek, átrendeznek, levágnak adott szabályok szerint és meghatározzák a származtatott adatok képzésének módját. Az eszközök egyes objektum előfordulás típusok azonosításához és kiválasztásához lehetnek specifikálva. További lehetőség, hogy a leképezések megfelelést adnak biztonsági és integritási szabályok között a séma vonatkozó szintjei között.

A leképző függvények a sémák konkrét tartalmától függetlenül készíthetők el, felhasználva az egyes szintek specifikus leképezés leíró nyelvét, amelynek utasításait a leképezés generáló processzorok dolgozzák fel. Ezeket a leképezéseket használhatjuk fel a cél-sémák generálásához. Másik lehetőség, hogy a leképezéseket a sémákkal együtt készítjük el, felhasználva egy közös séma/leképezés definiáló nyelvet, aminek feldolgozása során konzisztencia vizsgálatot is végezhetünk. A lefordított és kialakított "tárgy" leképezés könyvtárban tárolható.

Az egyes sémák objektumainak és a köztük lévő leképezések független deklarációja nemcsak arra jó, hogy minden adminisztrátor fenntarthassa egyedi, logikai szemléletét, hanem ami ennél fontosabb: a jól kia-

lakított leképző processzorok nagyobb flexibilitást engednek meg és a különböző szintű változásokat abszorbeálják anélkül, hogy más szintű leírásokat változtatni kellene. A változások nagy részéhez az érintett szintek közötti leképzés újradefiniálásával alkalmazkodhatunk.

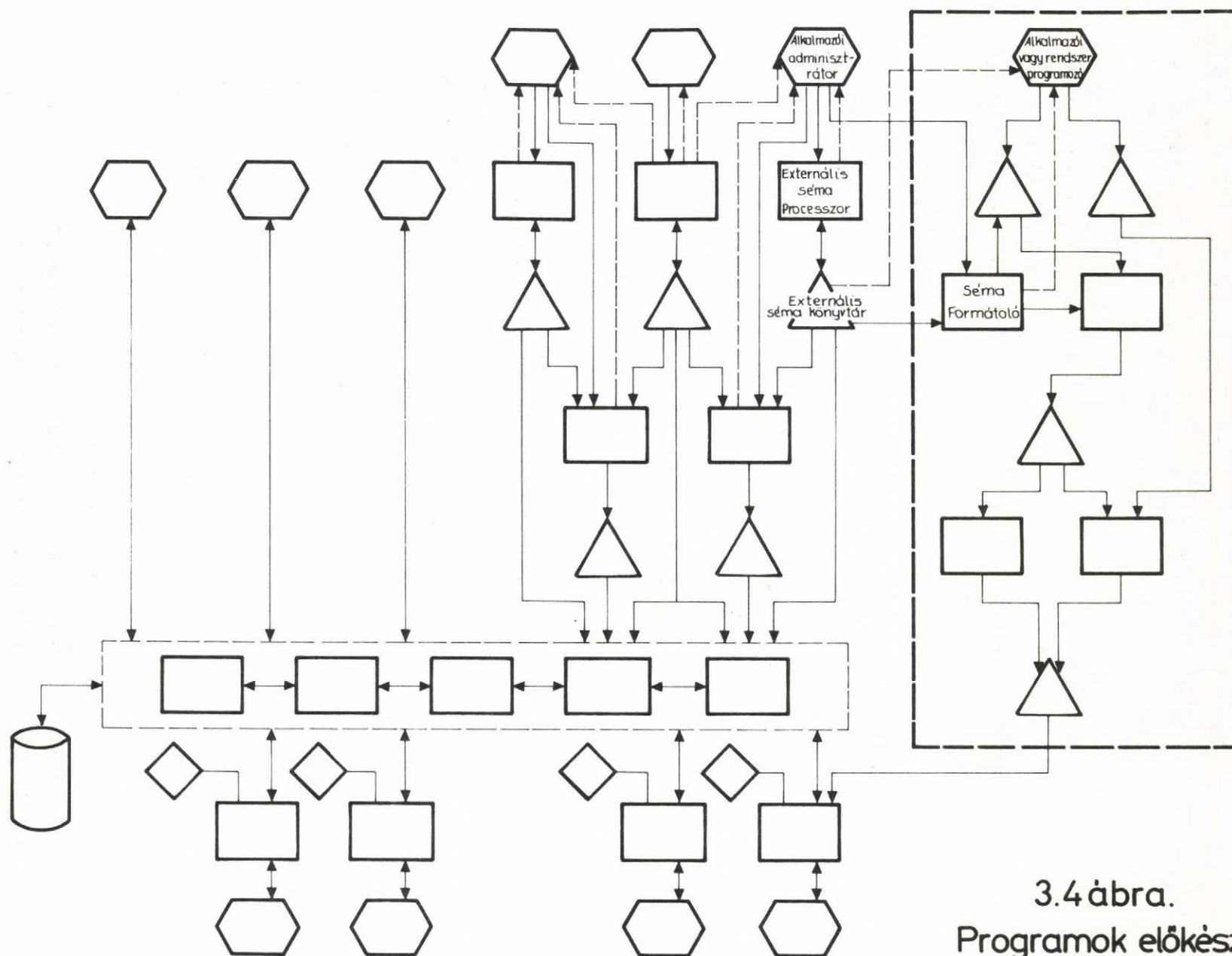
Lehetséges lenne az externális és internális objektumok között közvetlen leképzések megvalósítására is, amivel hatásosabb kiszolgálást kapnánk az externális kérések feldolgozásakor. Ezzel azonban veszítünk a koncepcionális séma által nyújtott adatfüggetlenségből. Hasonlóképpen leképzések lennének kialakíthatók közvetlenül az externális-internális sémák között úgy is, hogy az internális/koncepcionális sémák közötti, valamint a koncepcionális/externális sémák közötti leképzések szorzatát képezzük.

3.2 Program fejlesztés

3.2.1 Program elkészítés (3.4 ábra)

Az externális jellegű programok elkészíthetőségéhez az alkalmazói (vagy rendszer) programozóknak hozzáférést kell biztosítani a vonatkozó externális sémákhoz, amit ahhoz az alkalmazáshoz definiált valamelyik alkalmazói adminisztrátor. Az externális sémát megfelelő biztonsági védelem mellett átadják a programozónak vagy indirekt módon a séma formátumozón keresztül, vagy közvetlenül az externális séma forrásnyelvi változata formájában. Ennek birtokában a programozó képes a kívánt alkalmazáshoz illeszkedő program elkészítésére.

A programok nemcsak externális szinten írhatók, hanem az internális strukturák ismeretében is. A vállalati adminisztrátor döntésére van bízva, hogy



3.4 ábra.
Programok előkészítése

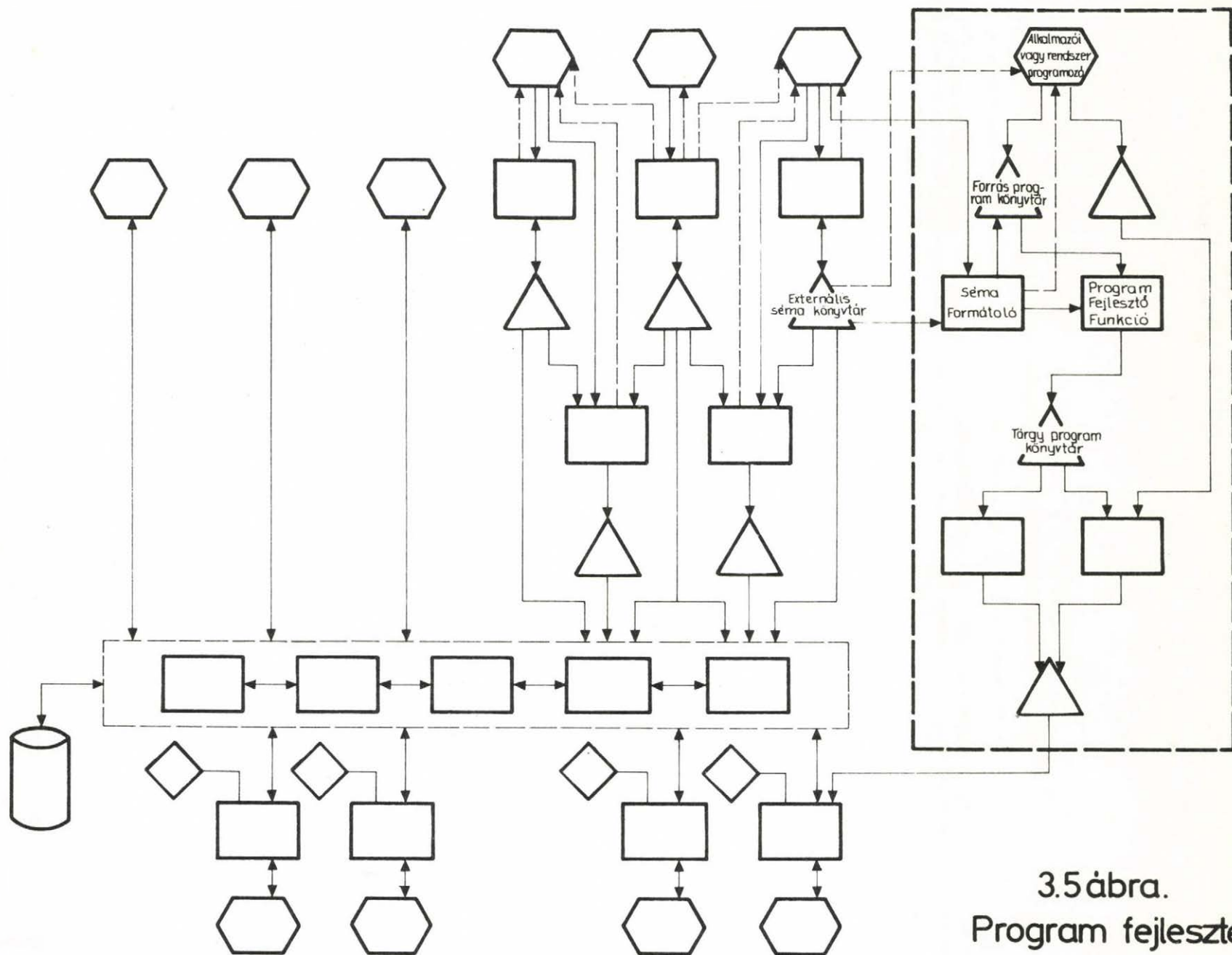
megtiltja-e az externálistól eltérő szintű programok működtetését, mivel ellenkező esetben az adatbázishoz való hozzáférési út elkerülhet magasabb szintű ellenőrzési pontokat, amelyeknél biztonsági/integritási információt tartalmaz a rendszer. Az externálistól eltérő programírás másik vetülete az adatfüggetlenség elvesztése vagy lényeges csökkentése, mivel bizonyos programokat közvetlenül olyan strukturákhoz kötünk ekkor, amiket alacsonyabb szintű interface definiál. Ezeknek nincs rugalmas alkalmazkodási képességük az előforduló változásokkal szemben.

Függetlenül attól, hogy a programot milyen szinten irták meg, annak forrásnyelvi változata tárolható forrásnyelvi program könyvtárban, ahonnan végrehajtás céljára előhívható.

3.2.2 Forrás program - tárgy program konverzió (3.5. ábra)

A program fejlesztési funkciót ellátó egység könyvtárból kapja a forrás programot, amit tárgy programmá konvertál. Eközben felhasználhatja valamelyik externális sémát, (amit a séma formátumozó egység átformátumozott) és összeköti az externális sémában szereplő neveket és objektum tulajdonságokat a programbeli utasításokkal ennél a pontnál. Ha az externális séma nem áll rendelkezésre a konverzió processzor munkájához, akkor az előbbiekben említett kötés az adatobjektumok és a program között később valósul meg.

A fordítás és kötés (részleges vagy teljes) után a tárgy program, tárgy program könyvtárban helyezhető el, ahonnan végrehajtáshoz előhívható.



3.5. ábra.
Program fejlesztés

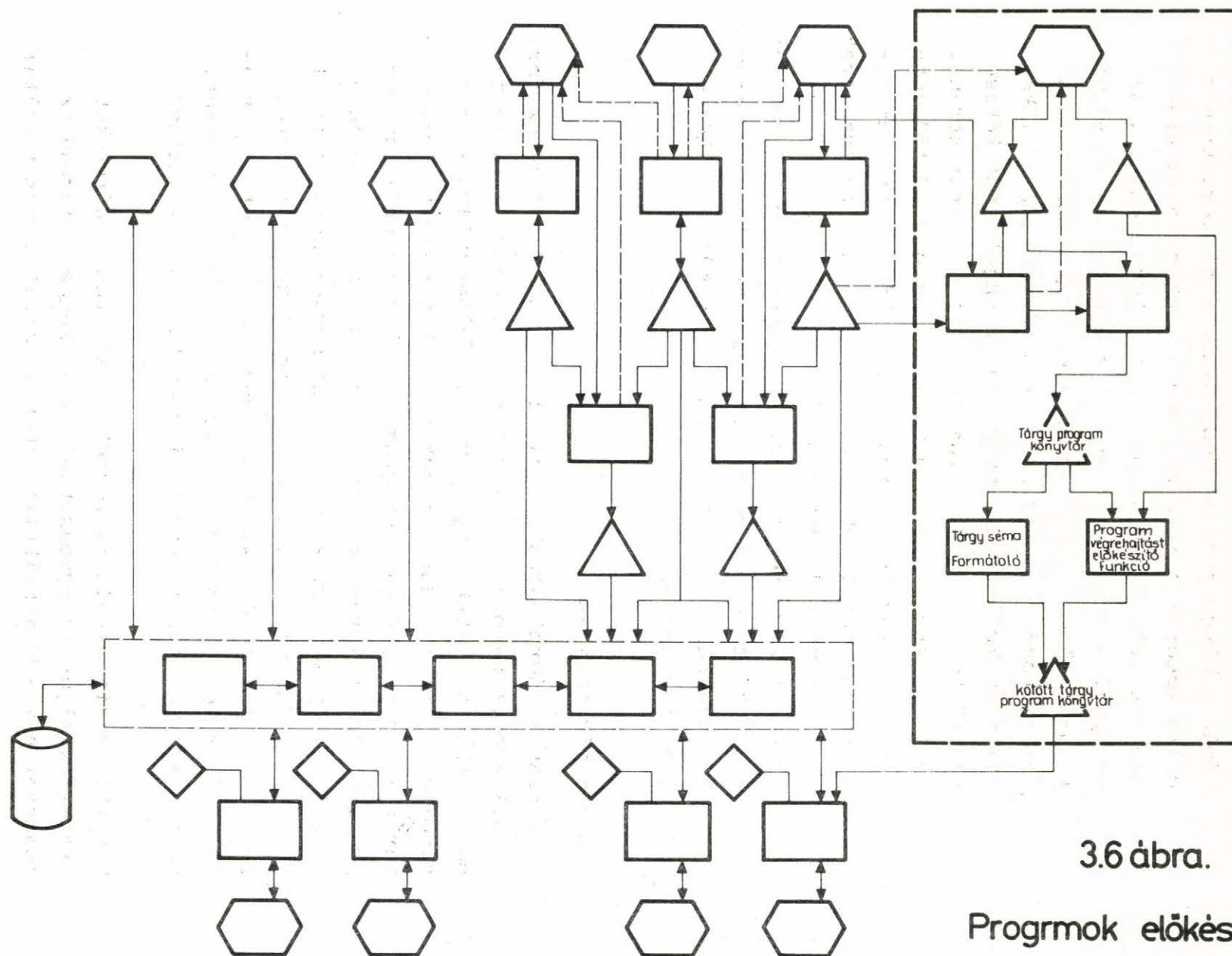
3.2.3 Végrehajtáshoz való előkészítés (3.6 ábra)

Ekkor kapcsolhatók össze egymással a tárgy program modulok, vagy a még teljesen nem kapcsolt externális sémák a tárgy program modulokhoz köthetők. Erre a célra speciális rendszer utasítások használhatók, amelyek irányítják az összekötés körülményeit. Az externális sémákban szereplő adatnevek és objektum tulajdonságok a tárgyprogramhoz akár a tárgy séma formátumozóval, akár a program végrehajtást előkészítő funkcióval köthetők (ha az objektumok kötése nem történt meg a program fejlesztő funkció során). Ha az illető alkalmazás magasabb szintű adatfüggetlenséget akar fenntartani (esetenként a program működési jellemzőinek rovására is), akkor az adatnevek és objektum tulajdonságok dinamikusan köthetők a programhoz végrehajtáskor. A teljes kötött vagy dinamikus kötéshez előkészített programok tárgy program könyvtárban tárolhatók, ahonnan betölthetők közvetlen végrehajtáshoz.

3.3 Program végrehajtás (3.7 ábra és 3.8 ábra)

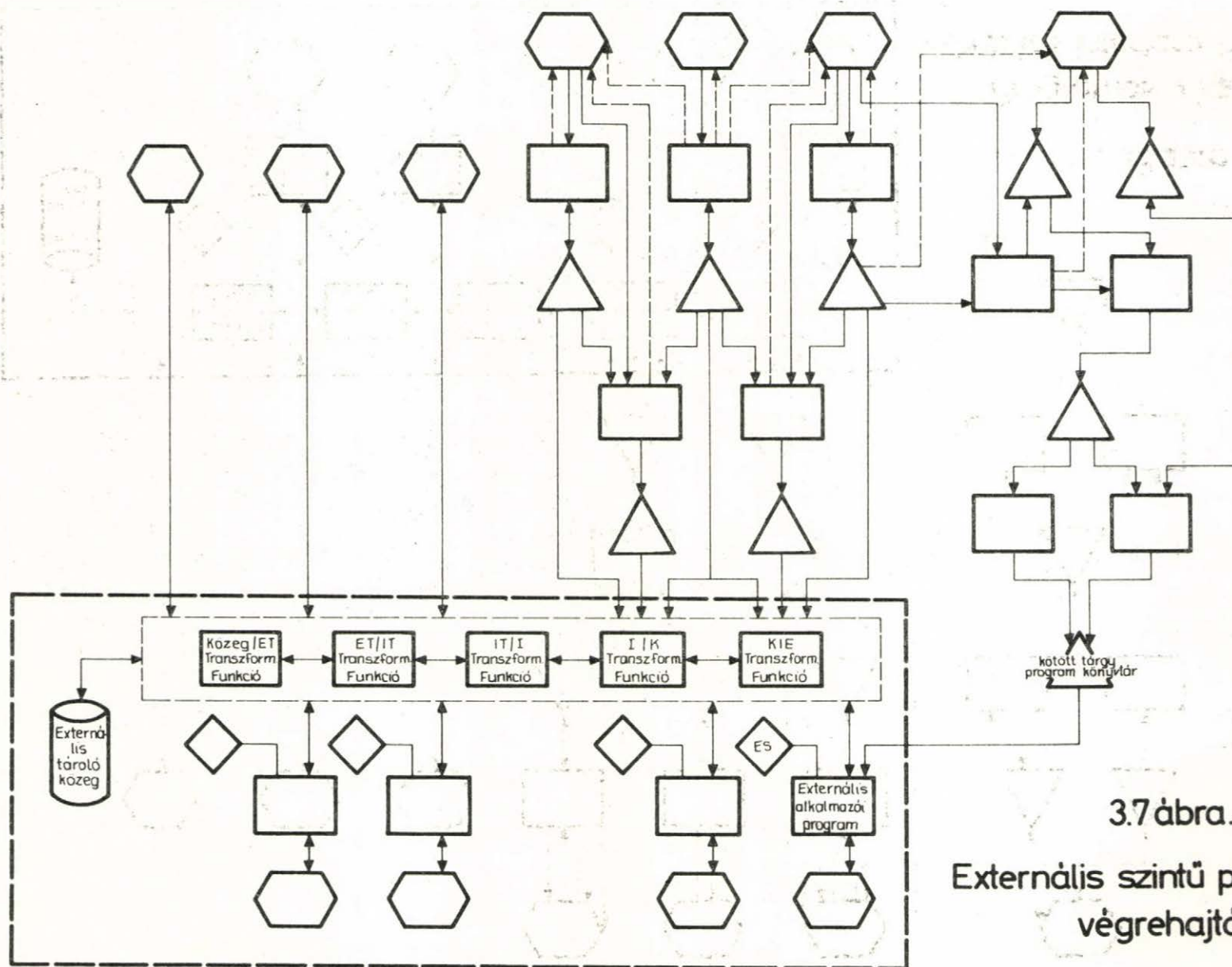
A programok végrehajtáskor általában az adatbázishoz fordulnak adatok tárolása, visszakeresése, módosítása céljából. Ezeket az ABKR hajtja végre felhasználva a programokban szereplő adatleírásokat, amiket vagy a sémakönyvtáron keresztül, vagy a programokból közvetlenül kap meg. Az illusztráló ábrák bemutatják hogyan transzformálódnak a kérések lépésenként az externális megjelenési formától az eszköz/közeg fizikai hozzáférésig. A konkrét adatbázis rendszer implementációk nem szükségképpen követik és realizálják különállóan ezeket a transzformációs lépéseket, hanem összevonásokat vagy áthidalásokat eszközölhetnek.

Miközben a kiszolgálási, végrehajtási uton a kéréseket transzformáljuk az egymásutáni szinteken, a sémákban az adatokat nem kell aktuálisan materializálni, transzformálni

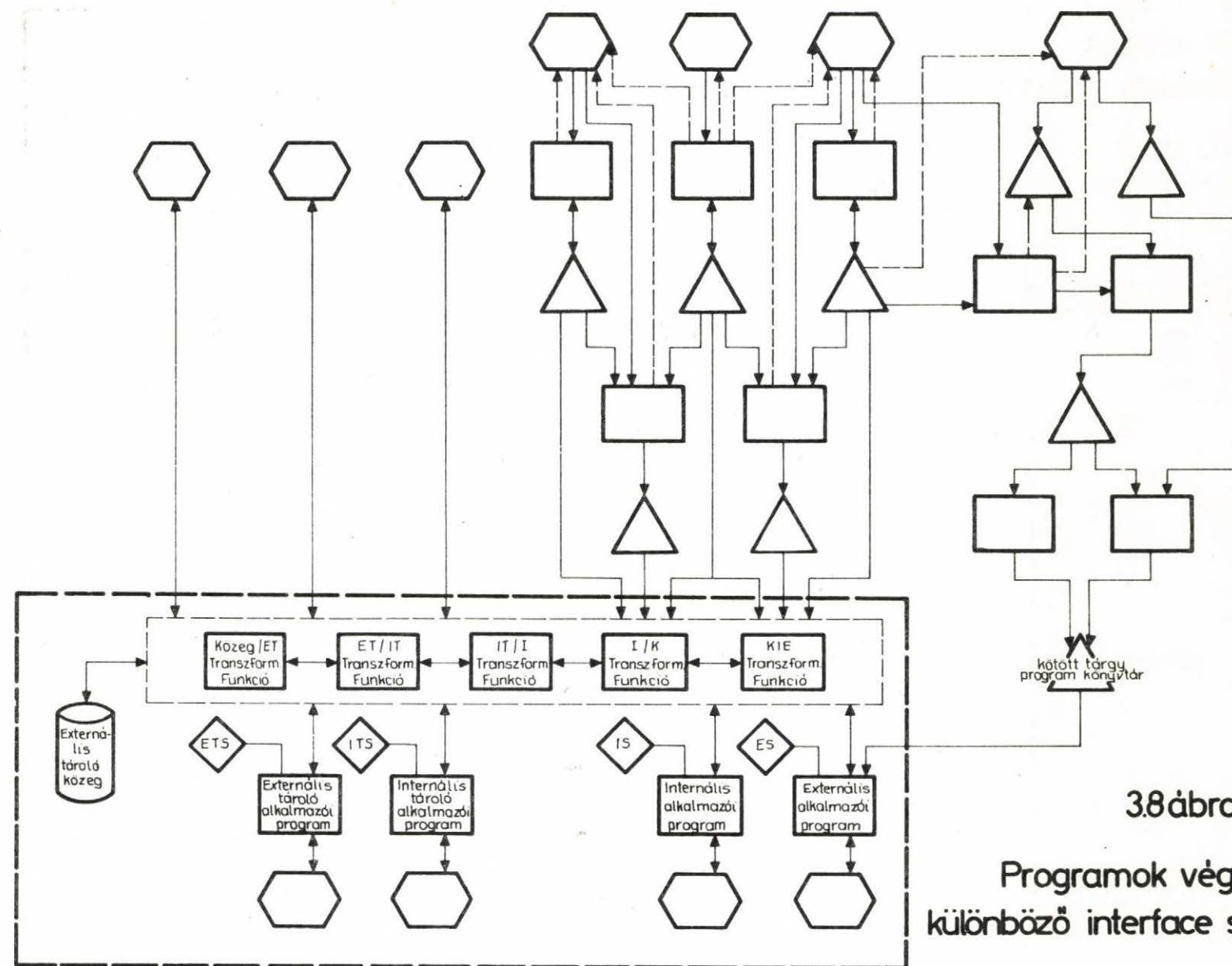


3.6 ábra.

Programok előkészítése
végrehajtáshoz



3.7 ábra.
Externális szintű programok
végrehajtása



3.8.ábra.

Programok végrehajtása
különböző interface szinteknél

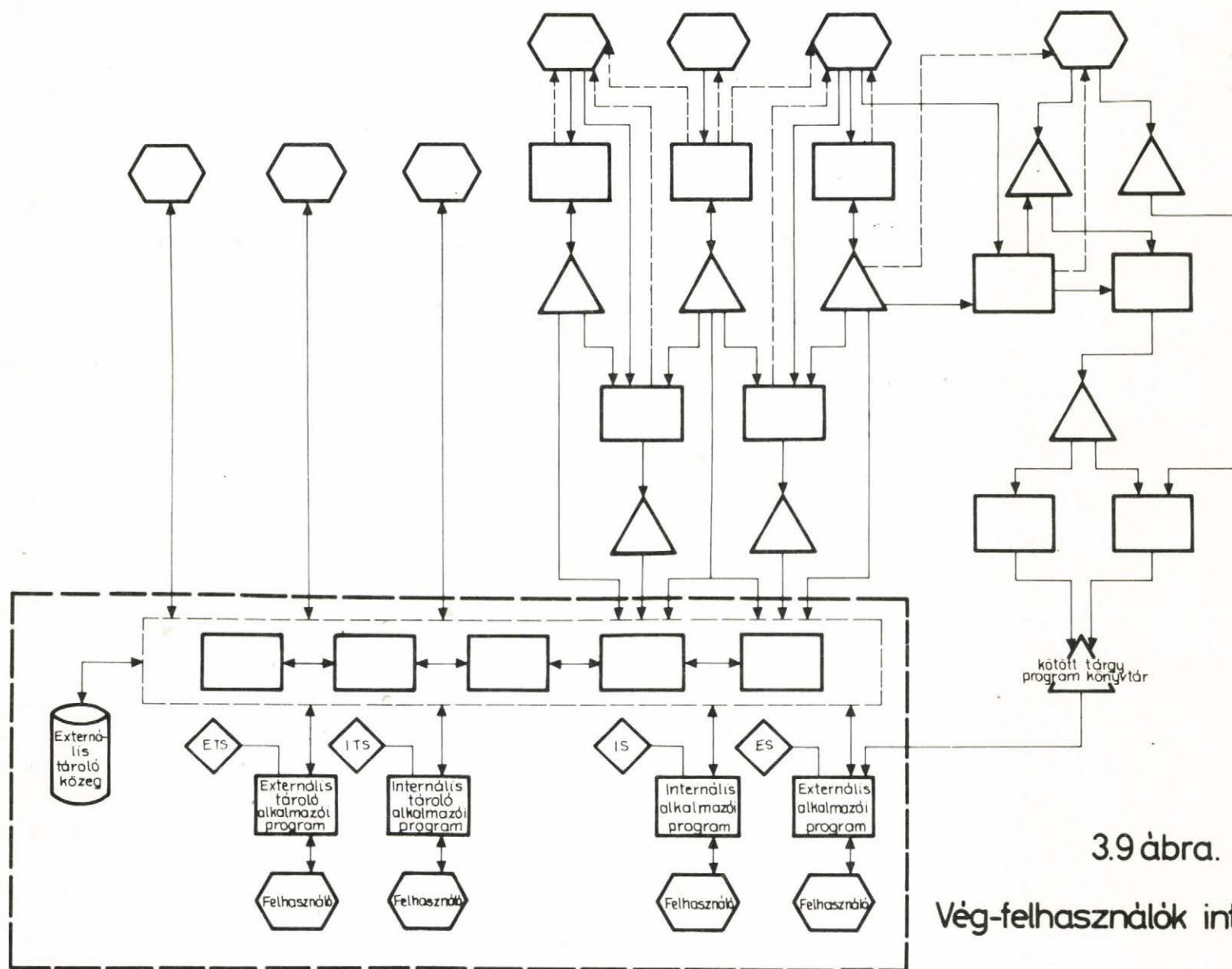
mindenütt. Például egy teljes visszakeresési művelet kiszolgálásakor a fizikai tároló közeg (diszk) valamely szektorából átvitel történhet egy internális tároló lapra, ahonnan a visszakeresett externális rekord előkészíthető a felhasználói program munkaterületén. Így miközben több transzformációs lépés során az érintett objektumok leíró jellemzőit végig pásztázza a vonatkozó transzformációs funkció az egyes interface felületeknél az adat objektumok tényleges materializálása csak két pontnál történik meg. (Az illusztráló ábrákon a hozzáférési funkció által érintett rendszer részek vannak feltüntetve).

3.4 Felhasználói interface-ok (3.9 ábra)

Az ABKR nagyon sokféle, eltérő koncepcióju felhasználót szolgál ki: reportok specifikálóit, lekérdező/felfrissítő akciók specifikálóit, parametikus felhasználókat (általában terminálon keresztüli hozzáférés) és az információs rendszer szolgáltatásait használó nem számítógépes szakembereket. Minden felhasználó egyedi, vagy általánosított felhasználói programon keresztül kommunikál a rendszerrel, amelyektől származó adatkérések az adatbázis rendszer különböző szintjeinél lévő, interface-knél kerülnek végrehajtásra. Ebben a rendszer filozófiában az egyes számítógép gyártók vagy más software irodák önmagában teljes adatbázis kezelő rendszerei speciális vagy általánosított alkalmazói programoknak tekinthetők, amelyek széles feladatosztályt alkalmazó orientált módon oldanak meg.

3.5 Az adatbázis fejlődése és karbantartása

A vállalati adminisztrátor folyamatosan figyeli és követi az információ hasznosítását a vállalaton belül, új információs igények jelentkezését és ennek megfelelően revideálja a koncepcionális sémát. Ebben a sémában történő változtatás általában módosítást eredményez az adatbázis környezetében is, azaz az egyes sémákat, leképzéseket, alkalma-



3.9 ábra.
Vég-felhasználók interface-ja

zói és rendszeren belüli programokat, leképzéseket, alkalmazói és rendszeren belüli programokat, másrészt az adatbázis tartalmát is változtatni kell.

Gyakoribb azonban, hogy a koncepcionális sémában rögzítésre kerülő, vállalaton belüli információ felhasználási mód viszonylag stabilis marad hosszabb időtartamra, viszont az internális és az externális környezet változtatást kíván. Ennek oka újabb hardware egységeknek a rendszerhez történő csatolása, újabb software rendszerek beillesztése, optimálisabb adatbázis struktúra kialakítása a hatásosság növelésére, újabb alkalmazói programoknak (vagy program rendszereknek) a többihez történő integrálása, adatbázis helyreállítása, átformátumozása, újra generálása lehet. A vállalatot jellemző koncepcionális séma érzékeny a vállalatot érintő lényeges változásokra, piaci körülményekre, új érdeklődési területek jelentkezésére, átszervezésekre, egyesülésekre és más dinamikus változásokra a vállalat környezetében. Ezen lehetőségek jelentkezése ellenére a koncepcionális séma stabilabb marad, mint a másik két séma. Így az internális és az externális sémákat a viszonylag stabil koncepcionális sémára támaszkodva hozzuk létre és nem egymásra alapozva. Ezzel az internális és externális sémákat oly módon szigeteljük el egymástól, hogy saját környezetükben annak sajátos adottságai miatt bekövetkező változások nem terjednek át a másik sémára és viszont. Olyan internális és externális séma processzorok leképzési mechanizmusok kialakításával, amelyek az internális és externális változásokat képesek követni, továbbá kellő szintű adatfüggetlenséget biztosítani az alkalmazói (valamint rendszer) programok számára elérhető a "változásokhoz alkalmazkodó" karakterű adatbázis rendszer.

Nyilvánvaló, hogy az ilyen alapokon nyugvó adatbázis modell nem oldja meg teljesen a változó környezet problémáját, vagy csak jelentős működési jellemző romlással tudja a kívánt szintű adatfüggetlenséget biztosítani. A koncepcionális sémát körülvevő keret azonban úgy van tervezve, hogy

a változásokhoz sokkal gyorsabban és gazdaságosabban alkalmazkodhassunk, mint az eddigi programozási eszközök birtokában képesek voltunk.

I R O D A L O M

- [1] ANSI/X3/SPARC Study Group, Interim Report. American National Standards Institute, February 1975.
- [2] T.B. Steel Jr. Data Base Standardization. A Status Reprot. Lecture Notes in Computer Science 39, Data Base Systems. Ed. by H. Hasselmeier and W.G. Spruth. Springer-Verlag Berlin-Heidelberg-New York 1976.
- [3] CODASYL Data Base Task Group Report, ACM, New York, 1971
- [4] Codd E.G. A Relational Model of Data for Large Shared Data Banks. CACM 13, 1970.
- [5] GUIDE/SHARE Data Base Management System Requirements SHARE Inc., New York, 1970.
- [6] CMSAG Joint Utilities Project Data Management Systems Requirement. CMSAG, Orlando, 1971.
- [7] Richter G. On the Relationship between Information and Data. Data Base Systems. Ed. by H. Hasselmeier and W.G. Spruth. Springer Verlag, 1976.
- [8] Sibley E.H. The Development of Data-Base Technology ACM Computing Surveys. Vol.8, No.1, March 1976.
- [9] Fry J.O, Sibley E.J. Evolution of Data - Base Management Systems. ACM (mint [8])
- [10] Chamberlin D.D. Relational Data-Base Management Systems ACM (mint [8])
- [11] Taylor R.W., Frank L.R. CODASYL Data-Base Management Systems ACM (mint[8])

- [12] Tsichritzis D.C., Lochovsky F.H. Hierarchical Data-Base Management
A Survey. ACM (mint [8])
- [13] Michaels A.S., Mittman B., Carlson C.R. A Comparison of the Relational
and CODASYL Approaches to Data-Base Management.
ACM (mint[8]).

